

論文96-33A-5-6

대용량 전전자교환기의 고속 내부통신망 성능 평가에 관한 연구

(A Study on the Performance Evaluation of High Speed Interprocessor Communication Network in a Large Capacity Digital Switching System)

崔眞圭*, 朴亨濬**, 鄭然快**, 權普燮**, 李忠根**

(Jin Kyu Choe, Hyeong Jun Park, Youn Kwaee Jeong, Bo Seob Kwon, and Chung Kun Lee)

요 약

본 논문에서는 대용량 전전자교환기의 새로운 고속 내부통신망(HIPC)에 대한 성능을 분석 평가하였다. HIPC는 변형된 라운드 로빈 중재방식으로 빠른 예약과 함께 중재가 이루어지는 구조이다.

HIPC의 성능 평가는 단일 서버에 다수의 큐를 갖고 비전면 서비스를 수행하는 시스템으로 해석적인 방법과 함께 시뮬레이션으로 수행하였다. 해석적 방법에 의한 결과와 시뮬레이션의 결과는 근사하게 일치하였다. 또한 HIPC에 TDX-10 기본 호처리절차를 적용하여 분석하였다. 그 결과를 기존 TDX-10 IPC와 비교하였으며, NTP 노드 발신버퍼에서의 평균대기시간의 차이가 메세지 도착율이 증가함에 따라 급격히 증가함을 알 수 있다.

Abstract

This paper presents performance evaluation of a new High speed Inter-Processor Communication(HIPC) network for large capacity and high performance digital switching system. The HIPC structure implements the fast reservation and concurrent arbitration technique(modified round-robin arbitration).

The performance evaluation of HIPC was performed by not only computer simulation but also numerical approximation method which was derived for a single server multi-queue system with nonexhaustive cyclic service. The approximation results are almost same with that of computer simulation. The TDX-10 basic call scenario was applied to the HIPC network and analyzed. These results were compared with TDX-10 IPC and shows that the difference of the mean waiting time in the TX buffer of NTP node increases sharply according to the increase of the message arrival rate.

* 正會員, 韓南大學校 電子工學科

(Dept. of Electronic Eng. Hannam Univ.)

** 正會員, 韓國電子通信研究所

(Electronics and Telecommunications Research Institute)

※ 본 논문은 한국전자통신연구소의 연구 지원으로 연구되었음.

接受日: 1995年3月27日, 수정완료일: 1996年3月23日

I. 서론

최근 교환기는 분산제어 방식이 보편화 되어있다. 분산제어 구조를 갖는 교환기의 제어계 구조를 살펴보면 교환기의 유지보수 및 운용과 관련된 기능들은 집중화되고 엄격한 실시간 처리가 요구되며, 반복성이 강한 기능들은 가능한 분산된 프로세서에서 분산처리하는 경향이 있다. 이 경우 호처리를 비롯한 교환기의 전반적인 기능을 효율적으로 수행하기 위해서는 분산된 프

로세서간 통신을 위한 기능이 매우 중요한 요소가 되어 이의 구조는 전체 시스템의 성능과 신뢰도, 앞으로의 시스템 확장성 등에 큰 영향을 미치게 된다.^[11]

국내에서 개발된 TDX-10 시스템은 최대 10만 가입자를 수용할 수 있는 대형 시스템으로서 제어계 구조는 다수의 프로세서가 완전 분산된 형태로 구현되어 있으며, 이들 프로세서간 통신은 별도의 고속 IPC(interprocessor communication) 네트워크를 통하여 수행된다. IPC 네트워크는 글로벌 버스에 기반을 둔 고속 메시지 스위치로 구현되어 있어 프로세서의 증설이 용이하며, 방송메세지의 전달 및 대용량 메세지의 송신 등이 가능한 장점을 지니고 있다.^[12] 반면에 시스템의 용량이 대용량화될 경우 시스템의 복잡도가 증가되어 가격 상승을 초래할 수 있는 원인이 된다. 따라서 앞으로 예상되어지는 다양한 서비스의 요구 증가를 쉽게 수용할 수 있으며 동시에 좋은 경쟁력과 경제성을 갖는 새로운 대용량 교환시스템에 필요한 대용량 고속 IPC 네트워크의 필요성이 절실히 대두되었다.

본 논문에서는 일반적인 대용량 고속 IPC 네트워크에 대한 요구를 충족시킬 수 있는 고성능 IPC 네트워크(HIPC : High speed IPC)의 구조와 성능을 분석하였다. HIPC 네트워크는 병렬 데이터버스와 기존 TDX-10 시스템에 적용된 단일 토큰 방식과 달리 두개의 토큰을 사용하는 변형된 round-robin 중재방식을 사용하여 버스의 이용율과 성능을 향상시켰다.^[13]

본 논문의 구성은 서론에 이어 2장에서는 기존의 TDX-10 IPC 네트워크의 구조와 HIPC 네트워크의 구조를 비교 분석하고, 3장에서는 HIPC 네트워크의 성능을 해석적 방법과 시뮬레이션을 통하여 평가하였으며, 동시에 TDX-10 기본 호처리 절차를 적용하여 기존의 TDX-10 IPC 네트워크와 비교 분석한 후 4장에서 결과를 종합 분석하여 정리하였다.

II. IPC 구조

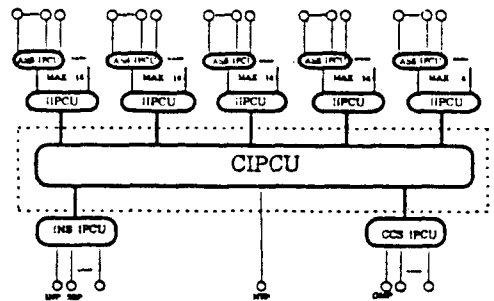
ISDN을 중심으로한 다양한 비음성 서비스의 증대와 이동통신 서비스의 수용으로 교환기의 프로세서간 통신 메세지는 기존 음성전용 교환기 환경하의 메세지보다 상당히 증가될 것으로 예상된다. 또한 이동통신 시스템을 고려하여볼 때 CIS(CDMA interconnect subsystem)를 중심으로한 BSC (base station controller)와 BTS(base station transceiver

subsystem)의 내부에는 프로세서간 통신뿐만 아니라 음성 데이터도 처리할 수 있는 대용량의 노드와 고속의 메세지 교환이 요구된다. 따라서 이러한 요구사항을 만족하고 교환기 주변 환경의 변화에 적응할 수 있는 통신 네트워크를 구현하기 위해서는 시스템 관점 하에서의 망 유닛 구조뿐만 아니라 단위 망을 구성하는 메세지 스위치의 효율적인 구성이 필요하다.

이를 위하여 HIPC는 TDX-10 IPC와 호환성을 유지하면서, 서브시스템간 3 레벨 성형 구조로 되어있는 기존 IPC네트워크를 2 레벨로 단순화시켜 프로세서간 고속 통신을 실현하였다. 이장에서는 TDX-10 IPC와 HIPC구조를 비교하고, HIPC에서의 새로운 버스 중재 방식을 설명하였다.

1. TDX-10 IPC 구조

그림 1은 계층적 구조를 갖는 TDX-10 IPC 구조이다.^[12] 여기서 ASS IPCU, INS IPCU, CCS IPCU, IIPCU, CIPCU는 10Mbps의 D-bus로 같은 하드웨어 구조이며, ASS IPCU와 IIPCU는 2Mbps fiber optic link로 연결되고, INS IPCU, CCS IPCU, IIPCU등과 CIPCU는 10Mbps의 U-link로 연결된다. 기타 PP, NTP, INP, SSP, OMP등 프로세서는 1Mbps(또는 2Mbps)의 U-link로 D-bus와 연결된다. D-bus는 최대 32개의 노드를 수용하는 고속의 버스로, 링크를 통하여 노드에 전달되는 메세지를 노드 순서에 따라 전송한다. ASS IPCU는 단말의 프로세서와



ASS : Access Switching Subsystem
 IPCU : Inter-Processor Communication Unit
 IIPCU : Inter IPCU
 CIPCU : Central IPCU
 INS : Inter-Network Subsystem
 CCS : Central Control Subsystem
 NTP : Number Transmission Processor

Physical U-link (Max. 10 MHz)
 Fiber optic link (Max. 2MHz)
 Processor U-link (1 MHz)

그림 1. TDX-10 IPC를 이용한 프로세서간 통신 네트워크

Fig. 1. Interprocessor communication network with TDX-10 IPC.

연결되며, 하나의 IIPCU에는 최대 14개의 ASS가 연결되어 전체적으로 60개의 ASS가 IIPCU를 통하여 CIPCU에 연결된다. INS IPCU와 CCS IPCU는 연결된 프로세서의 특성 상 각기 독자적인 IPCU를 통하여 CIPCU에 연결된다. INS IPCU에는 전화 연결을 위한 정보 및 처리를 하는 프로세서들로 하나의 INP와 SSP 8개가 연결된다. CCS IPCU에는 전화요금 등의 정보를 처리하는 프로세서인 OMP등이 연결된다. NTP는 전화번호를 번역하여 자국호인가를 판별하는 프로세서로서 교환되는 메시지 양을 고려하여 직접 CIPCU에 연결된다. D-bus 상 노드 버퍼의 크기는 8Kbyte이다.

2. HIPC 구조

그림 2는 개선된 HIPC의 구조이다. 이 구조는 단일 CIPCU로 통합된 2 레벨 IPC구조를 갖는다. 동일 버스 상에 최대 160노드 이상이 수용 가능해짐에 따라 TDX-10 IPC(그림1참조)의 IIPCU를 CIPCU로 통합하는 것이 가능해 졌으며 이에 따라 통신단계도 2단계가 축소되었다. 또한 기존의 데이터 버스는 시리얼 데이터(Serial data) 전송 방식을 이용하여 10Mbps의 속도로 구현되어 있었으나, 병렬 버스(Parallel Bus)구조(8개 라인)로 변경시켜 80Mbps 정도의 IPC 처리 용량을 갖게 되었다.

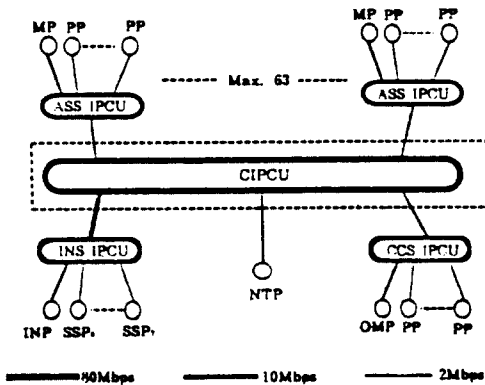


그림 2. HIPC를 이용한 프로세서간 통신 네트워크
Fig. 2. Interprocessor communication network with HIPC.

1) 버스 중재 방식

데이터 버스 사용 제어를 위한 중재 방식은 round-robin 방식을 바탕으로한 이중 토큰(실행 토큰, 예약 토큰) 운용 방식이 적용된다. 그림 3에서와 같이 토큰 예약 시스템은 어느 한 노드가 버스를 사용하기 시작할 때 전송할 메시지가 있는 다음 차례의 노드에

계 예약 토큰을 넘겨주어 버스 사용을 예약할 수 있게 한다. 버스 사용을 예약한 노드는 이전 서비스중인 노드의 수행 상태를 감시하고 있다가 서비스가 완료되면 실행 토큰을 동작시키고 다음 노드로 하여금 차례로 버스 사용을 예약할 수 있게 자신의 예약 토큰을 다음 노드로 이동시킨다. 예약 토큰을 동작시킬 수 있는 노드는 예약 토큰을 동작시킬 수 있는 기회가 주어지고 그 시점에서 데이터 버스로 송신할 프레임을 가지고 있는 노드이다.

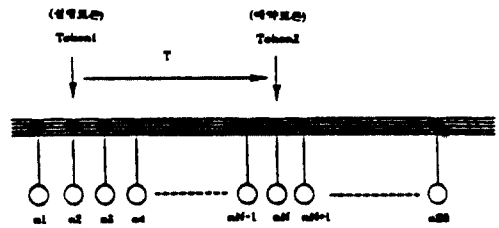


그림 3. 버스 중재 방식
Fig. 3. Bus arbitration.

III. 성능평가

본 장에서는 HIPC 구조를 다수의 큐와 단일 서버를 갖는 사이클릭 서비스 시스템으로 모델링하고 해석적 방법과 시뮬레이션을 사용하여 성능을 분석하고자 한다. 제한한 해석적 방법의 정확성을 시뮬레이션 수행결과와 비교하였으며, HIPC의 성능 평가 요소로는 노드 버퍼에서의 평균 메시지 대기 시간(average message waiting time) 및 버스 이용율(bus utilization)을 구하여, 그 결과를 TDX-10 IPC의 성능과 비교하였다. 또한 실제 호처리절차에 따른 메시지 입력 트래픽하에서 시뮬레이션을 행하여 호처리에 따른 성능 비교를 하였다.

1. 해석적 방법에 의한 성능분석

새로운 버스 중재 방식은 스위치오버 시간을 다음과 같이 나눌 수 있다. 먼저 토큰이 i 번째 큐에서 $i+1$ 번째 큐를 방문하는데 걸리는 시간을 아이들 스위치오버(idle switch-over) 시간이라 하고, i 번째 큐를 서버가 서비스한 후 그 다음 큐를 서비스하기 시작할 때까지(그다음 큐는 $i+1$ 번째 큐가 아닐수 있다) 걸리는 시간을 서비스 스위치오버 시간이라 한다. 새로운 방식을 정리하면 다음과 같다.

- 1) 기본적으로 사이클릭 서비스 방식과 같다. 그러나
- 2) 서버가 i 번째 큐를 서비스하는 도중에 토큰은 $i+1$ 번째 큐를 방문하고 메세지 유무를 체크한다. 없으면 $i+2, i+3, \dots$ 를 계속 체크한다. j 번째 ($i < j \leq \text{module}(i+N)$, 여기서 N 은 시스템에 있는 큐의 갯수) 큐에 메세지가 있으면 그 큐에 머무른다.
- 3) 서버가 i 번째 큐를 서비스 완료하면 서비스 스위치 오버 시간을 경과한 후 토큰이 머문 큐를 방문해 서비스를 시작한다.
- 4) 서버가 i 번째 큐를 서비스 완료한 후에도 토큰이 머물지 못하면 (이 경우 토큰이 각 큐를 계속적으로 방문하고 있는 중이다) 토큰이 머무를 때까지 서버는 기다린다.
- 5) 토큰이 머물고 있는 큐에 서버가 방문하면, 그 즉시 서버는 서비스를 시작하고 토큰은 그 다음 큐를 방문한다.

사이클릭 서비스 방식은 pseudo-conservation law^[9]를 이용해 대기시간을 구한다. 그 외 Kuehn^[5]은 조건 사이클 시간을 도입해 대기시간을 구했으나 시뮬레이션과 많은 차이가 있다. 본 논문에서는 Boxma^[11]가 분석한 개념을 기초로 해 대기시간을 구하였다.

메세지들은 각각 $\lambda_1, \lambda_2, \dots, \lambda_N$ 을 갖는 독립적인 포와송 프로세스(independent Poisson process)로 각 큐에 도착하며, 메세지 서비스 시간은 i.i.d.(independent, identically, distributed)하며 그 평균(혹은 first moment)과 second moment 값은 $\beta_i, \beta_i^{(2)}$ ($1 \leq i \leq N$)을 갖는다. 또한 서비스 프로세스는 도착 프로세스(arrival process)와 스위치오버 프로세스와는 무관(independent)하다.

예약 토큰이 없다고 가정하면 사이클릭 서비스 방식에 대한 Boxma와 Meister의 해석법^[11]을 이용하여 큐 i 에서의 메세지 평균 대기시간 Ew_i 는 다음 식으로 구할 수 있다.

$$Ew_i \approx \frac{1-\rho+\rho_i}{1-\rho-\lambda_i S} \cdot \frac{1}{\sum_{j=1}^N (\rho_j(1-\lambda_j E[c_j]) \cdot \frac{1-\rho+\rho_j}{1-\rho-\lambda_j S})} \cdot \left[\frac{\rho}{2(1-\rho)} \sum_{j=1}^N \lambda_j \beta_j^{(2)} + \frac{\rho}{2S} \sum_{j=1}^N \phi^2 + \frac{S}{2(1-\rho)} \sum_{j=1}^N \rho_j (1+\rho_j) \right] \quad (1)$$

여기서 $\rho_i = \lambda_i \beta_i, (1 \leq i \leq N),$
 $\rho = \sum_{i=1}^N \rho_i = \sum_{i=1}^N \lambda_i \beta_i, (1 \leq i \leq N)$

$$S = \sum_{i=1}^N s_i, E[c] = S + \sum_{i=1}^N \lambda_i E[c] \beta_i,$$

$\phi^{(2)}$ = 큐 i 에서 $(i+1)$ 까지의 아이들 스위치오버 시간의 분산.

스위치오버시간 중 서비스 스위치오버 시간은 임의의 큐에 메세지를 서비스할 때만 존재하므로 서비스 시간에 포함시킬 수 있다. 즉 임의의 큐 i 에서의 서비스 시간 β_i 는 서비스 스위치오버 시간 s_i 을 합하여, 새로운 서비스 시간 β_i' 으로 볼 수 있다.

$$\beta_i' = \beta_i + s_i, \beta_i'^{(2)} = \beta_i^{(2)} + s_i^{(2)} \quad (2)$$

(2)식을 기초로 해, 각 큐 i 의 사용율(utilization)과 서버의 전체 사용율(total utilization)을 다음과 같이 수정한다.

$$\rho_i' = \lambda_i (\beta_i + s_i) \quad 1 \leq i \leq N \quad (3)$$

$$\rho' = \sum_{i=1}^N \rho_i' = \sum_{i=1}^N \lambda_i (\beta_i + s_i) \quad (4)$$

아이들 스위치오버시간 s_i' 는 i 번째 큐에서 $i+1$ 번째 큐로 토큰이 방문하는데 걸리는 시간이므로 i 큐에서 j 만큼 앞에 있는 큐부터 영향을 받는다. $(i-j)$ 번째 큐가 메세지를 서비스하고 그 사이의 큐들에서는 모두 서비스할려는 메세지가 없으면 s_i' 은 이 서비스 시간에 흡수된다. 즉 $s_i'' = 0$ 이다. 또, $(i-j)$ 번째 큐와 i 번째 큐 사이에 서비스해야 하는 메세지가 적어도 하나 이상이면 s_i' 는 서버가 서비스하는 시간에 흡수된다. 즉 s_i' 이 존재하려면 $(i-j)$ 번째 큐부터 i 번째 큐 사이에 전송할려는 메세지가 없어야 한다. 여기서 j 는 $(i-j)$ 번째 있는 메세지의 서비스 시간 $\beta_{(i-j)}$ 를 $s_{(i-j)}''$ 으로 나눈값이고 각 큐들의 서비스 시간이 같다면, 즉 $\beta_i = \beta (1 \leq i \leq N)$ 이면 $j = \frac{\beta}{s_i}$ 이다. 또한 제안된 방식에서 s_i' 는 상수이므로

$$j = \frac{\beta_{(i-j)}}{s_{(i-j)}} = \frac{\beta}{s_i}$$

가 된다.

가정 1. 임의의 큐 i 에서 토큰이 그 큐를 방문할 때 큐가 비어있을 확률은

$$P_i(0) = 1 - \lambda_i E[c] \quad \text{이다.}$$

여기서 $\lambda_i E[c]$ 는 임의의 큐 i 에 도착하는 메시지의 평균 개수이고, 한 사이클동안 임의의 큐 i 에 도착하는 메시지의 평균수는 큐에서 서비스를 받고 나가는

메시지의 평균수와 같다.¹⁵⁾

가정 1을 사용하면 한 사이클 시간동안 전체의 스위치오버 시간 S 와 서버가 한 사이클을 서비스하는데 걸리는 평균시간 $E[c]$ 의 관계를 나타낼 수 있다.

$$E[s_i] = (1 - \lambda_i E[c])^{\min[N, \frac{\beta_i}{s_i}]} \cdot s_i \quad (5)$$

$$S = \sum_{i=1}^N E[s_i] = \sum_{i=1}^N (1 - \lambda_i E[c])^{\min[N, \frac{\beta_i}{s_i}]} \cdot s_i \quad (6)$$

$$E[c] = S + \sum_{i=1}^N \lambda_i E[c] (\beta_i + s_i) \quad (7)$$

(5)식을 Taylor 급수의 2차항까지 보면

$$E[s_i] \approx (1 - \min[N, \frac{\beta_i}{s_i}] \cdot \lambda_i E[c]) s_i \quad (8)$$

$$S \approx \sum_{i=1}^N (1 - \min[N, \frac{\beta_i}{s_i}] \cdot \lambda_i E[c]) s_i \quad (9)$$

$$E[c] = \sum_{i=1}^N s_i - \sum_{i=1}^N \min[N, \frac{\beta_i}{s_i}] \cdot \lambda_i E[c] \cdot s_i + \sum_{i=1}^N \lambda_i E[c] (\beta_i + s_i) \quad (10)$$

이 된다.

따라서 식(1)에 식(2),(3),(4),(9),(10)을 대입하여, 새로운 방식에서 임의의 큐 i 에 대한 메시지 평균대기시간 Ew_i 를 구할 수 있다.

$$Ew_i \approx \frac{1 - \rho_i + \rho_i}{1 - \rho - \lambda_i S} \cdot \frac{1}{\sum [\rho_i (1 - \lambda_i E[c]) \cdot \frac{1 - \rho_i + \rho_i}{1 - \rho - \lambda_i S}]}$$

$$\left[\frac{\rho}{2(1-\rho)} \sum_{i=1}^N \lambda_i \beta_i^{(2)} + \frac{\rho}{2S} \sum_{i=1}^N \psi^2 + \frac{S}{2(1-\rho)} \sum_{i=1}^N \rho_i (1 + \rho_i) \right] \quad (11)$$

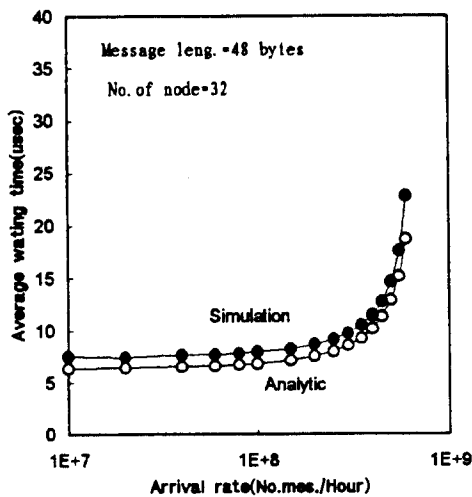


그림 4. 해석적 모델과 시뮬레이션 결과 비교
Fig. 4. Comparison of analytic model and simulation results.

위 식(11)를 이용하여 메시지 도착율(메세지 수/시간)에 따른 각 노드에서의 평균 메세지 대기시간을 구하였다. 구한 값을 검증하는 방법으로 HIPC 내부통신망의 동작을 모델링하여 시뮬레이션한 결과와 비교하였다.

그림 4는 메세지 길이가 48바이트이고 32노드인 경우, 해석적 방법과 시뮬레이션에 의한 결과치를 비교하였다. 해석적인 방법에 의하여 구한 결과와 시뮬레이션의 결과가 거의 근사하게 일치하여 해석적 방법으로 구한 결과가 타당함을 알 수 있다.

2. 기존 TDX-10 IPC와 HIPC와의 성능 비교

앞에서 구한 시뮬레이션 결과를 이용하여 TDX-10 IPC와 HIPC의 성능을 비교하였다. 그림 5는 기존 TDX-10 IPC와 HIPC와의 평균 메세지 대기 시간을 비교한 것이다. 메세지 길이가 48바이트일 때, HIPC의 메세지 대기시간은 기존 TDX-10 IPC에 비해 시간당 메세지 도착율이 1.0×10^7 일 때 1/4, 4.0×10^7 일 때는 1/8, 8.0×10^7 일 때는 1/50 정도가 되는 것을 알 수 있다. 따라서 메세지 도착율의 증가에 따른 메세지 대기시간의 변화량은 기존 TDX-10 IPC에 비해 매우 적어 HIPC의 고속성을 알 수 있다. 또한 시간당 메세지 도착율을 비교해보면 메세지 대기시간이 40 μ s가 되는 도착율은 기존 TDX-10 IPC 경우 2.0×10^7 정도이지만 HIPC는 6.5×10^8 정도로 약 33배가 되어 HIPC의 대용량성을 보여주고 있다.

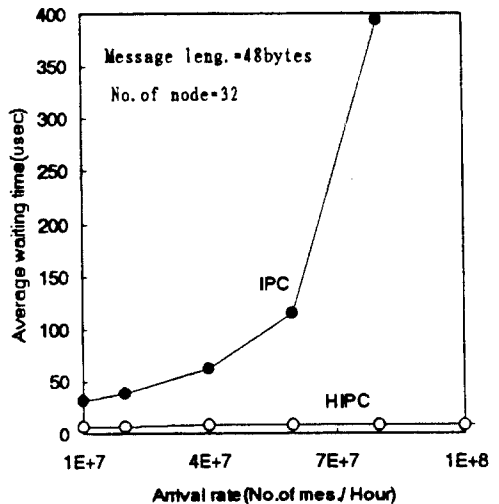


그림 5. 평균 메세지 대기시간 비교
Fig. 5. Comparison of average message waiting times.

그림 6은 기존 TDX-10 IPC의 버스 속도를 HIPC와 같이 80Mbps로 가정하였을 때 각 노드에서의 평균 대기시간을 비교한 것으로 시간당 메시지 도착율이 작을 때는 큰 차이가 없으나, 도착율이 커질수록 차이가 커지는 것을 알 수 있다. 이것은 HIPC의 새로운 버스 중재방식인 이중토큰방식에서 예약토큰에 의한 스위치 오버시간의 감소에 의한 것으로 분석되며, 이중토큰방식에 의하여 성능이 개선됨을 나타내고 있다.

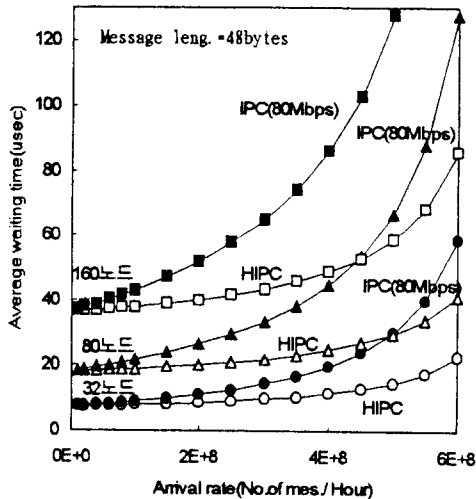


그림 6. 평균 메시지 대기시간 비교
Fig. 6. Comparison of average message waiting times.

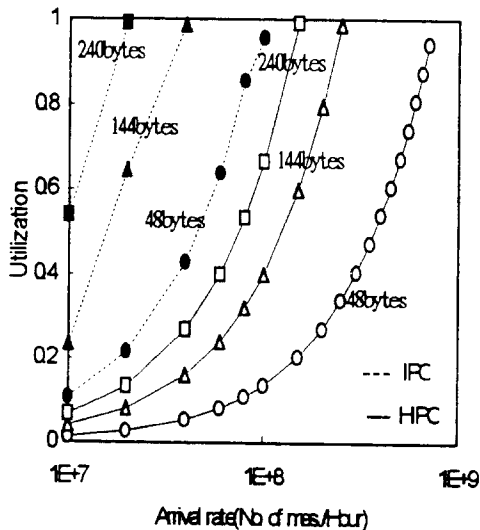


그림 7. 버스이용율 비교
Fig. 7. Comparison of bus utilization

그림 7은 메시지 길이변화에 따른 버스 이용율의 비교이다. 메시지 길이가 48바이트일 때 메시지 도착율을 비교해보면, 기존 TDX-10 IPC에 비하여 HIPC가 약 8배정도 향상된 성능을 보인다.

3. TDX-10 호처리절차를 적용한 성능분석

대용량교환기는 LCS(Local Call Simulator)와 UCS(Universal Call Simulator)등을 이용하여 시스템에 요구되는 최대 용량의 트래픽을 가한 후 성능을 입증하기에는 현실적으로 불가능하다. 따라서 실제적인 상황에서의 성능을 분석하기 위하여 TDX-10 교환기에서 사용되는 기본 호처리절차에 따라 메시지가 도착하는 가상상황을 설정하고 이를 시뮬레이션 하였다. 성능 평가 요소로는 노드버퍼에서의 평균 메시지 대기시간 및 버스 이용율을 중심으로 수행하였으며 그 결과를 TDX-10 IPC의 성능과 비교하여 특성을 분석하였다. 시뮬레이션 언어로는 SLAM II를 사용하였다.¹²⁾

1) 시뮬레이션 모델

내부통신망을 통해 프로세서 상호간에 교환되는 정보는 ①호처리에 관련된 정보, ②초기적재(initial loading)시 프로그램, ③시스템 동작에 관련된 정보, ④시스템 유지에 관련된 정보, ⑤정보메세지 등으로 구분할 수 있다.⁶⁾ 이중 ②, ③, ④, ⑤의 정보는 호처리와 무관하거나 정확한 평가가 어려우므로 성능 분석에서 제외한다. 따라서 호처리에 관련된 정보가 내부통신망 분석을 위한 메시지를 이루게된다고 가정한다. 호들은 내부통신망 프로토콜(protocol)과 호처리절차에 따라 일정 개수의 메시지를 각각 생성하며, 이중 특정 호에 관련된 메시지들은 상호 종속적이다. 각 호의 첫 메시지들은 시간적으로 poisson분포에 따라 내부통신망에 도착하리라고 가정한다. 이들 메시지의 길이는 프로토콜과 호처리절차에 따라 주어진다.

내부통신망은 그림 1에서 정의한 바와 같이 모델링하였다. 모두 63개의 ASS가 2Mbps 링크를 통하여 새로운 중재방식의 80Mbps의 CIPCU에 연결되어 메시지를 전송한다. INS IPCU는 10Mbps, CCS (Central Control Subsystem) IPCU는 2Mbps, NTP는 2Mbps로 CIPCU와 연결된다. INS IPCU는 새로운 중재방식을 사용하는 10Mbps Dbus이며, INP는 2Mbps 링크로 연결되며, 나머지 8개의 SSP들은 1.25Mbps로 연결된다. 노드와 노드사이 전환시간은 단순 스위치오버 시간인 경우는 0.4μsec이며, 예약노드

로의 서비스 스위치오버 시간은 0.2μsec이다. 즉, 후속 노드에메세지가 없는 경우 다음에 메세지가 도착하는 노드로의 스위치오버 시간은 0.4μsec이며, 새로운 중재 방식에 따라 하나의 메세지가 전송되고 있는 동안 진행되던 예약 토큰에 의해 미리 예약된 노드로의 스위치오버시간은 0.2μsec가 걸린다.

Poisson분포로 도착되는 각 호들은 기존 TDx-10의 호처리절차에 따라 각 프로세서와 메세지를 교환하게된다. 이때 호에 의해 발생된 메세지의 길이는 기존 TDx-10 IPC에서 정의된 평균길이를 사용한다.^[2] 발생된 모든 호는 연결이 완전히 이루어지는 완전호인 것으로 가정한다. CCS IPCU의 OMP로는 5번의 호가 완료된 ASS에서 256바이트의 메세지가 전송된다. 노드에서 링크를 통하여 프로세서에 메세지를 전송할 경우 메세지 전송 후 96개의 전송클럭 만큼의 지연이 추가된 후에 새로운 전송을 개시할 수 있다. 시뮬레이션 프로그램은 SLAM II의 이산사건 모델을 이용하여 작성하였다.

2) 시뮬레이션 결과

내부통신망 성능을 평가하는 요소로 트래픽 변화에 따르는 노드버퍼의 변화를 중심으로 버퍼에서의 대기 시간을 구하였다. 버퍼대기시간은 총서비스시간에 관련되므로, HIPC의 처리능력을 평가하는 중요 척도가 된다.

부가되는 트래픽은 시간당 메세지 도착수(도착율)로 시간당 100만개의 메세지가 도착하는 도착율에서 시간당 500만개의 메세지가 도착하는 도착율까지 변화시키면서 시뮬레이션 하였다. 이 도착율이상은 NTP와 CIPCU사이의 링크용량의 한계로 더 이상 시뮬레이션 할 수 없었다. 즉 NTP와 CIPCU사이의 링크속도가 전체 시뮬레이션의 병목현상을 초래하여 더 이상의 시뮬레이션이 의미가 없게 된다.(시간당 500만개의 메세지 도착율에 이르면 NTP의 수신 링크의 이용율이 1에 달하게 되어,그 이상 도착율을 증가시키더라도 NTP 수신노드버퍼 처리 속도의 한계로 전체 성능에 왜곡이 일어나게된다.) 이 경우 비교대상이 된 TDx-10 IPC의 링크속도도 HIPC와 마찬가지로 2Mbps로 하여 비교하였다.

시뮬레이션 결과 HIPC의 전송속도가 매우 높아서 전체적으로 TDx-10 IPC에 비하여 우수하게 나타났다. TDx-10 IPC와 HIPC는 메세지전송을 위한 발신 노드버퍼에 대기시간은 많은 차이를 보이며, CIPCU이

용율 또한 많은 차이를 보인다. 그림 8은 TDx-10 IPC와 HIPC의 이용율의 비교를 보인다. 여기서 HIPC는 도착율에 상당한 여유가 있어서 더 많은 트래픽을 처리할 수 있음을 알 수 있다. 따라서 HIPC는 앞으로 더 많은 트래픽이 예상되는 이동통신이나 ISDN등에도 무리없이 사용될 수 있을 것이다.

그림 9는 메세지가 발생된 ASS에서 NTP까지의 지연시간에 대한 비교이다. 여기서도 HIPC는 TDx-10 IPC에 비하여 월등한 성능을 보이고 있다. 500만 정도의 도착율로는 커다란 지연효과가 없었다. 그림 10은 NTP 발신노드버퍼에서의 지연을 보인다. 이 경우 HIPC와 TDx-10 IPC는 도착율 증가에 따라 큰 차이를 보이고 있다. 그림 11은 INS에서의 INP발신노드버퍼의 지연시간을 보인다. INS DBUS는 TDx-10 IPC와 HIPC 모두 10Mbps의 전송속도를 갖지만, HIPC에는 새로운 중재방식이 적용되었다. 이 경우 HIPC와 TDx-10 IPC는 큰 차이를 보이고 있다. 도착율 증가에 따라 TDx-10 IPC는 급격히 지연시간이 증가하지만, HIPC는 지연시간의 증가가 매우 미미한 편이다.

수신노드 버퍼는 수신링크속도에 의하여 큰 영향을 받게 되는데, TDx-10 IPC와 HIPC가 동일한 링크속도를 사용하므로 지연시간이나 버퍼길이변화에 큰 차이가 없다. 그림 12는 NTP 수신노드 버퍼에서의 평균 대기시간이다. 이 경우 도착율이 증가함에 따라 거의 같아진다는 것을 보여주고있다.

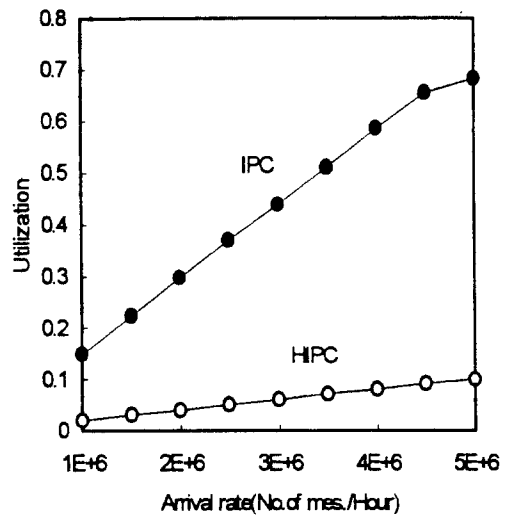


그림 8. CIPCU 이용율 비교
Fig. 8. Comparison of CIPCU utilization.

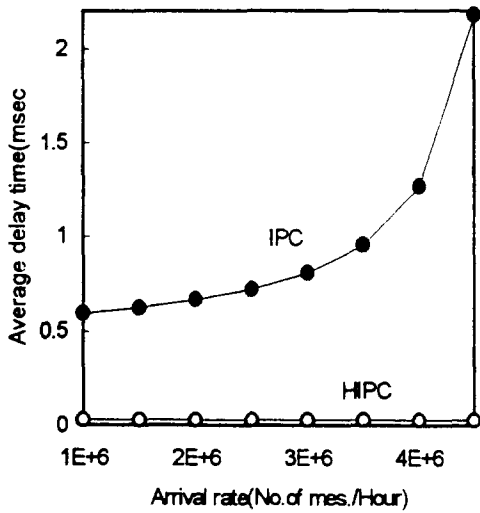


그림 9. ASS발신노드에서 NTP수신노드까지 평균지연시간
 Fig. 9. Comparison of average delay time from ASS Tx node to NTP Rx node.

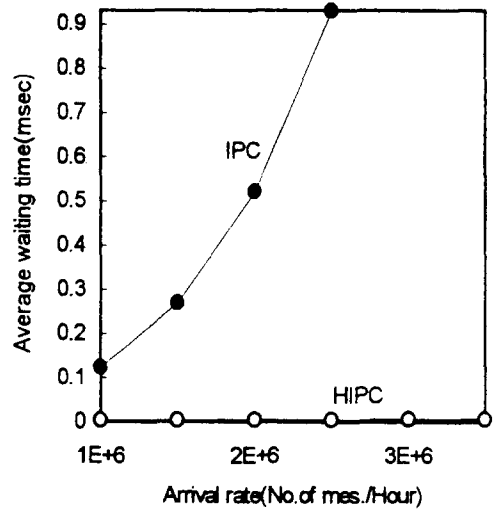


그림 11. INP발신노드 평균대기시간
 Fig. 11. Comparison of average waiting time for INP Tx node.

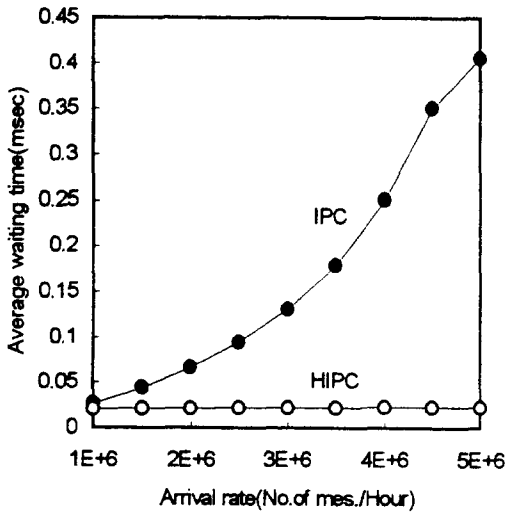


그림 10. NTP발신노드 평균대기시간
 Fig. 10. Comparison of average waiting time for NTP Tx node.

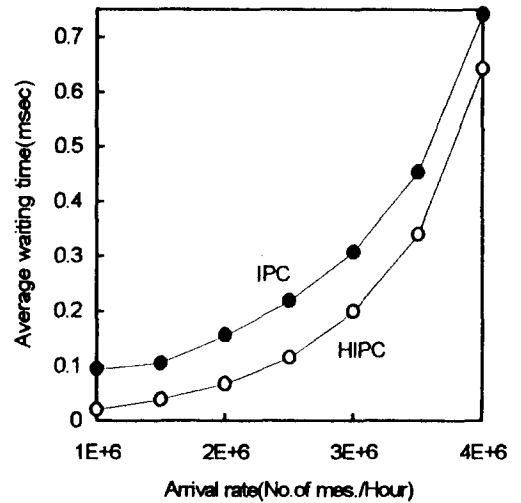


그림 12. NTP수신노드 평균대기시간
 Fig. 12. Comparison of average waiting time for NTP Rx node.

IV. 결론

본 논문에서는 새로운 고속 IPC 네트워크인 HIPC에 대하여 해석적인 방법과 시뮬레이션으로 성능과 특성을 분석하고, 기존의 TDx-10 IPC와 비교하였다. 또한 실제적인 상황에서의 성능을 평가하기 위한 방법

으로 TDx-10의 기본 호처리절차를 적용한 시뮬레이션으로 HIPC와 기존의 TDx-10 IPC의 성능을 비교 분석하였다.

단일 서버에 다수의 큐를 갖고 있고 비전면 서비스 정책을 갖고 있는 큐잉 모델에 새로운 버스 중재방식을 적용하여 HIPC의 성능을 분석할 수 있는 근사식을 구하였고, 이 해석적인 방법에 의해 얻어진 값을 시뮬레이션에 의해 구한 값과 비교한 결과 근사하게 일치

함을 보여 해석적 방법에 의한 성능 분석이 가능함을 보였다.

TDX-10 IPC와 HIPC와의 성능 분석 비교 결과는 메시지 길이가 48바이트일 때, HIPC의 메시지 대기시간은 기존 TDX-10 IPC에 비해 시간당 메시지 도착율이 1.0×10^7 일 때 1/4, 4.0×10^7 일 때는 1/8, 8.0×10^7 일 때는 1/50 정도가 되어 HIPC의 고속성을 알 수 있었다. 시간당 메시지 도착율을 비교해보면 메시지 대기시간이 $40\mu s$ 가 되는 도착율이 기존 TDX-10 IPC 경우 2.0×10^7 정도이지만 HIPC는 6.5×10^8 정도로 약 33배가 되어 HIPC의 대용량성을 보여주고 있다.

또한 기존 TDX-10 IPC의 버스 속도를 HIPC와 같이 80Mbps로 가정한 후 평균대기시간을 비교하였을 때 시간당 메시지 도착율이 작을 때는 큰 차이가 없으나, 도착율이 커질수록 차이가 커지는 것을 알 수 있었다. 이것은 HIPC의 새로운 버스 중재방식인 이중토콘 방식에 의한 성능 개선을 알 수 있다.

TDX-10 기본호처리절차를 적용한 시뮬레이션 결과에서 HIPC는 시간당 500만개 정도의 도착율로는 커다란 지연시간의 차이가 없으며, 버퍼의 대기시간에서도 HIPC는 새로운 중재방식이 적용되어 지연시간이 매우 적어지게 됨을 보이고 있다.

이상의 연구에서 HIPC는 노드수가 증가하여도 우수한 성능을 보임으로 고속의 데이터 전송능력과 대용량성 및 확장성이 있음을 알 수 있다. 또한 HIPC는 도착율에 상당한 여유가 있어서 더 많은 트래픽을 처리할 수 있으므로, HIPC는 앞으로 더 많은 트래픽이 예상되는 이동통신용 대용량 교환시스템이나 ISDN등에도 무리없이 사용될 수 있을 것이다. 본 연구에서 분석 평가된 성능은 내부통신망의 개선 및 새로운 방식 연구에 유용한 자료가 되며, 평가 방법은 앞으로의 연구에 많은 도움을 줄 수 있을 것이다.

참 고 문 헌

[1] L.G. Anderson, C.H. Bowers, D.L. Carney,

J.J. Kulzer, "Distributed systems trade-offs," vol. 1 of 4, *Proceeding of International Switching Symposium*, Mar. 1987.

- [2] TDX-10 시스템 구조, 한국전자통신연구소, 1991.
- [3] 김상훈, 박형준외, "TDX-10에서의 고속 대용량 IPC 기능 구현", 정보과학회 학술발표대회, 1993
- [4] L.Kleinrock, *Queueing system*, vol.I, John Wiley & Son, 1975.
- [5] P.J.Kuehn, "Multi-queue systems with nonexhaustive cycle service," *Bell Sys. Tech. Journal*, Mar. 1979.
- [6] M.J.Ferguson and Y.J.Aminetzah, "Exact result for nonsymmetric token-ring systems," *IEEE Trans. on Comm.*, COM-33, pp.223-231, Mar.1985.
- [7] K.S.Watson, "Performance evaluation of cyclic service strategies - a survey," *Performance '84*, North Holland, pp. 521-533, 1984.
- [8] S.W.Fuhrmann, "Symmetric queues served in a cycle order," *Opr. Res. Letters*, vol.4, no.3, pp.139-144, Oct. 1985.
- [9] O.J.Boxma and W.P.Groenendijk, "Pseudo-conservation laws in cycle - service system," *Journal of Appl. Prob.*, vol.24, 1987.
- [10] S.W.Fuhrmann and Y.T.Wang, "Mean waiting time approximation of cyclic service systems with limited service," *Performance '87*, pp.253-265, 1987.
- [11] O.J.Boxma, B.W.Meister, "Waiting-time approximation for cycle-service systems with switchover times," *Performance '86*, pp.254-261, 1986.
- [12] A.Alan B.Pritsker, *Introduction to Simulation and SLAM II (3ed)*, Systems Publishing Corporation, 1986.

저 자 소 개



崔眞圭(正會員)

1958년 9월 20일생. 1980년 2월 고려대학교 전자공학과 졸업(공학사). 1982년 2월 고려대학교 대학원 전자공학과 졸업(공학석사). 1987년 8월 고려대학교 대학원 전자공학과 졸업(공학박사). 1987년 9월 ~ 1990년 8월 대전산업대학교 전자계산학과 조교수. 1990년 9월 ~ 현재 한남대학교 전자공학과 부교수. 주관심분야는 교환시스템, Interconnection network, ATM, 병렬처리, 컴퓨터구조 등임



朴亨濬(正會員)

1961년 5월 26일생. 1987년 2월 한양대학교 공과대학 전자공학과(공학사). 1987년 2월 ~ 현재 한국전자통신연구소 선임연구원. 주관심분야는 고속통신망, ATM 스위치망, VLSI CAD 등임.



鄭然快(正會員)

1959년 12월 15일생. 1982년 2월 경북대학교 공과대학 전자공학과(공학사). 1984년 2월 경북대학교 대학원 전자공학과(공학석사). 1993년 2월 경북대학교 대학원 컴퓨터공학과 박사과정 수료. 1983년 3월 ~ 현재 한국전자통신연구소 교환기술연구단 선임연구원. 주관심분야는 실시간 고속통신망, 이동통신시스템, 대용량 교환시스템 및 ATM 스위치 등임



權普燮(正會員)

1958년 7월 4일생. 1983년 2월 경북대학교 공과대학 전자공학과(공학사). 1990년 2월 충남대학교 대학원 전자공학과(공학석사). 1992년 ~ 현재 한국과학기술원 전산학과 박사과정 재학중. 주관심분야는 ATM Switch Architecture, Parallel Computer Architecture, Interconnection Network 등임



李忠根(正會員)

1951년 12월 4일생. 1976년 2월 고려대학교 전자공학과 졸업(공학사). 1989년 9월 고려대학교 대학원 전자공학과 졸업(공학석사). 1994년 8월 고려대학교 대학원 전자공학과 졸업(공학박사). 1978년 5월 ~ 1983년 4월 삼성전자(주). 1983년 6월 ~ 현재 한국전자통신연구소 이동통신기술연구단 이동통신망 연구부장. 주관심분야는 교환기, 이동통신, 광대역통신 등임