

論文96-33A-11-4

ISDN 사설교환기의 결합내성 구현 및 성능평가

(The Implementation and Performance Evaluation of the Fault-Tolerance for an ISDN PABX)

崔在洵*, 朴仁甲**

(Jae Weon Choe and In Kap Park)

요 약

본 논문에서는 교환시스템에 예기치 못한 이상이 발생하더라도 사용자의 음성이나 데이터 통신서비스를 계속 유지시키는 신뢰성 있는 ISDN 사설교환기를 구현하고자 하였다. 결합내성의 구현을 위해 ISDN 사설교환기의 주요 제어부분을 동작계열과 대기계열로 이중화하여 동작계열에 치명적 오류가 발생하게 되면 대기계열이 계열변경을 수행하여 새로운 동작계열로서 진행중이던 음성이나 데이터 서비스를 계속 진행할 수 있도록 하였다. 결과적으로 시스템에 어떤 이상이 발생하더라도 사용자는 결합을 전혀 느끼지 못한 상태에서 음성이나 데이터 통신서비스를 제공받을 수 있음을 성능평가를 통하여 입증하였다.

Abstract

In this paper we wanted to implement a reliable ISDN PABX that could maintain the user's voice and data communication services, even when there might be an unexpected system error. In order to implement the fault-tolerance we made this ISDN PABX composed of an Active and a Standby side by duplicating the main control parts. Whenever a critical error occurs in the active side, it makes side exchange so that the standby side as a new active side continues to proceed the currently active voice or data services. Finally, through the performance evaluation we proved that user's voice or data communication can be serviced without any sense in spite of a system error.

I. 서 론

교환기는 정보교환을 위한 매개체로서 정보화 시대를 이끌어 가는 중추적 역할을 해 왔다. 그런데 최근 서비스의 다양화로 교환시스템의 복잡성이 급속도로 증가함에 따라 이의 신뢰성 문제가 그 어느때 보다 중요하게 부각되게 되었다. 시스템의 신뢰성 향상을 위한 구현방법은 여러가지가 있겠지만 오류발생을 사전에

완전히 배제하기란 실제로 거의 불가능하므로 오류의 발생시 대응하는 결합내성(fault tolerance) 방법으로 교환시스템의 신뢰성을 향상시키는 연구가 진행되고 있다^[1].

일반적으로 교환시스템의 결합내성을 위하여 시스템의 중요 요소를 중복시켜 하나의 요소에 오류가 발생하더라도 중복된 다른 요소가 이를 대체하여 서비스를 계속 유지시키는 이중화 방식으로 구현하는 경향이 많다. 이중화 방식은 양측을 동시에 구동하여 명령어나 기능 단위로 동기를 맞추고 장애 발생시 최대한 빨리 복구 가능하도록 하는 동기식 이중화와 어느 한 쪽만이 동작하다가 장애 발생시 다른 한 쪽이 동작을 이어가는 비동기식 이중화로 대별된다^[2]. 동기식 이중화의 경우 수행단위별 동기화가 요구되므로 교환기와 같이 병렬처리를 지원해야 하는 대형 시스템에는 구현이 어

* 正會員, 三星電子 情報通信研究所
(Telecommunication R&D Center, Samsung Electronics)

** 正會員, 建國大學校 電子工學科
(Dept. of Elec. Eng., Kon Kuk University)

接受日字: 1996年4月15日, 수정완료일: 1996年10月29日

려워 부적합하다. 비동기식 이중화의 경우에는 어떻게 양측의 데이터를 동일하게 유지시킬 것인지, 또 장애시 얼마나 빨리 계열변경하여 서비스의 연속성을 보장할 것인지가 중요하다. 국설교환기 TDX-10의 이중화는 고속의 하드웨어 채널을 이용하여 동작계열의 처리결과를 메모리에 쓸 때 동시에 대기계열의 메모리에 씌으로써 계열간에 데이터의 동일성을 유지하고, 계열변경시 대기계열은 동작계열의 상태(스택포인트, 프로그램카운터 등)를 넘겨받아 수행을 계속하는 비동기식 이중화로 구현되어 있다^[3]. 그러므로 하드웨어 오류에 대해선 동기식 이중화와 같은 정도의 서비스 연속성을 보장하나, 소프트웨어 오류에 있어선 오류로 인해 깨진 데이터가 대기계열의 메모리에도 함께 쓰여져 오류의 누적으로 동작계열이 다운되면 대기계열 역시 다운되기 때문에 서비스의 연속성 보장에 상당한 문제를 내포하고 있다.

당 연구소에서는 ISDN 시대를 대비하여 일정 가입자 용량의 노드가 최대 8 개까지 망구조로 연결되어 최대 12000 가입자를 지원하고, 분산구조, 계층구조, 모듈구조 및 분산제어의 특징을 가지며 음성과 데이터 통신서비스를 제공하는 ISDN 사설교환기(INFOREX)^[4]를 개발하였다. 본 논문에서는 이의 신뢰성 향상을 위해 각 노드별로 주요 제어부분을 동작계열과 대기계열로 이중화(duplication)하고, 비동기식 이중화방식에 의거하여 결함내성을 구현함으로써 교환기의 어느 특정 하드웨어나 소프트웨어 모듈에 결함이 발생하더라도 전체 시스템 다운을 유발하지 않도록 하였다. 또한 호처리를 위한 필수 제어데이터만을 소실없이 정확히 전달하고, 계열변경후 넘겨받은 제어데이터를 가지고 재 동작하게 함으로써 오류파급에 의한 동시 다운현상이 발생하지 않도록 하였다. 그리하여 궁극적으로 시스템에 어떤 이상이 발생하더라도 사용자측에선 이러한 결함을 전혀 느끼지 못한 상태에서 진행중이던 음성 혹은 데이터 통신서비스를 계속해서 제공받을 수 있도록 하고자 하였다. 이러한 연구목표를 수행하기 위해 우선 요구사항을 분석하여 설계하고, 제안한 구현방법에 의거하여 구현한 후, 이의 성능을 평가하여 서비스의 연속성이 유지됨을 입증하였다.

II. 결함내성 기능

교환기의 신뢰성이란 고장없이 주어진 기능을 수행할

수 있는 능력의 측도를 말하며, 이를 높이기 위한 접근 방법은 크게 결함예방(fault prevention)과 결함내성(fault tolerance)으로 구분되어 진다^[1]. 기존의 결함 예방은 고품질의 소자를 사용하여 최악의 경우를 대비한 시스템을 구현함으로써 사전에 발생가능한 모든 오류를 제거하는 방법이다. 이와는 달리 결함내성은 표준품질의 소자를 사용하여 시스템을 구현하기 때문에 오류의 발생 가능성은 내재하지만, 부가적 소자를 사용하여 중복화함으로써 오류가 발생시 대처하는 방법이다.

결함내성 방법은 시스템의 중복화로 추가적 비용이 요구되므로 기존의 많은 상용 시스템이 결함예방 방법을 채택하여 시스템을 신뢰성을 향상시켰지만, 사전에 가능한 모든 오류를 고려한 완벽한 시스템을 구현한다는 것은 실재론 거의 불가능하다. 또한 결함내성 방법은 어느 정도의 오류 발생을 허용한 후 이의 발생시 대처하기 때문에 결함예방과 같은 시스템의 완벽성은 요구되지 않으므로 표준품질의 소자로 시스템을 구현할 수 있지만, 결함예방 방법은 오류 발생을 사전에 완전히 배제하기 위해 고품질의 소자가 요구되므로 비용 측면에 있어서도 결코 무시할 수 없다. 즉, 결함예방은 오류 대응력에 한계가 있고 비용이 많이 드는 단점이 있어 교환시스템의 신뢰성 향상을 위한 접근방법으로 결함내성이 적합하다.

결함내성 시스템(fault-tolerant system)이란 시스템의 동작중 어떤 범주의 오류가 발생할 경우 외부의 도움없이 정상 수행을 지속할 수 있는 역량을 가진 시스템을 말한다. 이의 구현을 위해서는 시스템의 수행에 필수적인 요소를 중복적으로 구성하여 해당 요소에 이상이 발생하더라도 시스템이 연속적으로 수행할 수 있도록 중복화(redundancy)해야 한다. 중복화는 여러 모듈을 동시에 동작시켜 나온 수행결과의 대표값을 취함으로써 결함모듈의 오류 영향을 제거시키는 동기식 중복화와 여러 모듈로 구성되어 있지만 특정 시점에는 하나만이 동작하고 오류 발생시 다른 모듈이 이의 수행을 대체하는 비동기식 중복화로 나뉜다^[2]. 그런데 동기식 중복화를 이용하여 수행결과의 대표값을 취하기 위해선 최소 3 개의 동일 모듈이 요구되므로 많은 비용이 요구되고, 매 오퍼레이션 마다 수행결과의 동기화가 요구되므로 교환기와 같은 대형 시스템에는 부적합하다. 그리고 중복화로 인한 비용 증대를 고려한다면 시스템의 오류발생에 대비하여 단지 하나의 모듈만을 두는 것이 바람직하므로 비동기식 이중화 방식으로 본

교환시스템의 결합내성을 구현하도록 한다^[5-6]. 또한 호처리를 위한 필수 제어데이터만을 소실없이 정확히 전달하고, 계열변경후 넘겨받은 제어데이터를 가지고 재 동작하게 함으로써 오류과급에 의한 동시 다운현상이 발생하지 않도록 한다.

시스템을 동작계열과 대기계열로 이중화시 동작계열은 실제로 시스템을 제어하면서 사용자에게 서비스를 제공하고 대기계열은 동작계열의 이상에 대비하여 동작하도록 한다. 이를 위해 동작계열은 평상시에는 대기계열의 정상 동작여부를 주기적으로 점검하고 계열변경에 필요한 각종 정보를 수시로 대기계열로 보내어 계열 상호간에 동일상태를 유지할 수 있도록 하고, 대기계열은 계열변경에 필요한 각종 정보를 동작계열로부터 수시로 받아 동작계열과 동일상태를 유지시킨다. 그리고 동작계열의 수행중 시스템에 치명적 오류가 발생하게 되면 계열변경 명령을 내려 대기계열이 계열변경과 복구작업을 거쳐 새로운 동작계열로서 수행을 계속하도록 만든다. 그리하여 현재 서비스가 진행중인 가입자는 이러한 시스템 이상을 전혀 인식하지 못한 상태에서 계속 서비스를 받을 수 있도록 한다.

교환기는 시스템의 진단이나 감시기능에 의해 시스템을 지속적으로 점검하여 기 정의한 하드웨어 오류나 소프트웨어 오류^[5]가 발생하면 이중화 타스크를 구동하여 계열변경과 복구를 수행한다. 또한 사용자 시스템간 통신기능을 통해 시스템 운용자가 강제적 계열변경이나 다양한 이중화 관련기능을 수행 가능하도록 한다^[5,7].

III. 결합내성의 설계

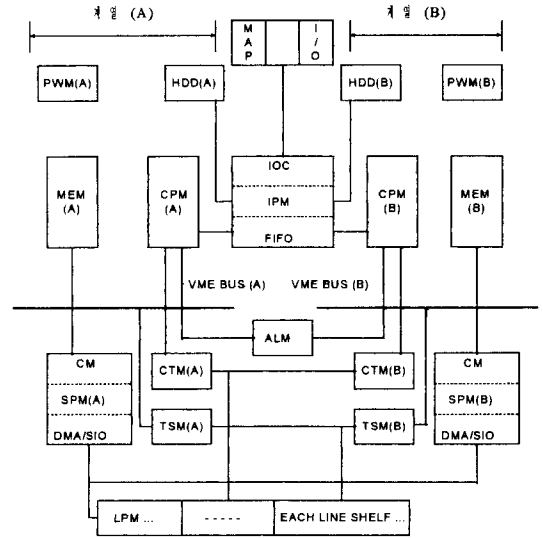
교환기가 결합내성을 갖도록 하기 위해 이중화된 교환기의 하드웨어와 소프트웨어 구조, 정보 교환을 위한 인터페이스 구조, 이중화 기능을 보조하는 각종 타스크와 모듈의 기능과 상호간의 관계를 설계하였다.

1. 이중화의 하드웨어 구조

교환시스템을 그림 1과 같이 각 노드별로 주요 제어부분을 이중화하여 어떤 부분에 예기치 못한 이상이 발생하더라도 시스템이 정상적으로 동작하도록 설계하였다^[8].

2. 이중화의 소프트웨어 구조

교환시스템의 이중화를 위한 소프트웨어(이하 이중화 타스크라 함)의 내부구조는 그림 2와 같다.



중앙처리모듈(CPM), 신호처리모듈(SPM), 임출력모듈(IPM), 타임스위치모듈(TSM), 메모리모듈(MEM), 클럭톤모듈(CTM), 가입자처리모듈(LPM), 파워모듈(PWM), 하드디스크(HDD)

그림 1. 이중화된 ISDN 사설교환기의 하드웨어 구조
Fig. 1. Hardware structure of Duplicated ISDN PABX.

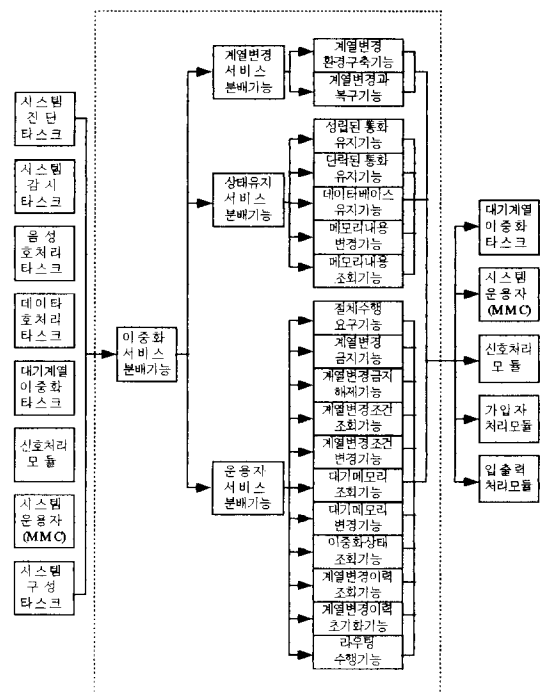


그림 2. 이중화된 ISDN 사설교환기의 소프트웨어 구조
Fig. 2. Software structure of Duplicated ISDN PABX.

3. 이중화의 인터페이스 구조

본 교환시스템의 이중화를 위한 인터페이스 구조는 그림 3과 같고, 실시간 운영체제의 다양한 시스템호출 기능을 이용하여 각종 제어정보를 교환한다. 계열상호간에 동일상태를 유지하기 위해 동작계열은 평상시에 FIFOTX2 시스템호출을 이용하여 FIFO2 큐를 통해 각종 제어정보를 대기계열과 교환한다. 시스템 이상시에는 대기계열이 새로운 동작계열로서 수행하도록 하기 위해 CMTX 시스템호출을 이용하여 공통메모리를 통해 계열변경 메시지나 재구동 메시지를 신호처리 모듈로 내려 보낸다. 또한 시스템 운용자가 이중화 관련 부가기능의 수행과 그 결과를 출력하기 위해 FIFOTX1 시스템호출을 이용하여 FIFO1 큐를 통해 통신한다.

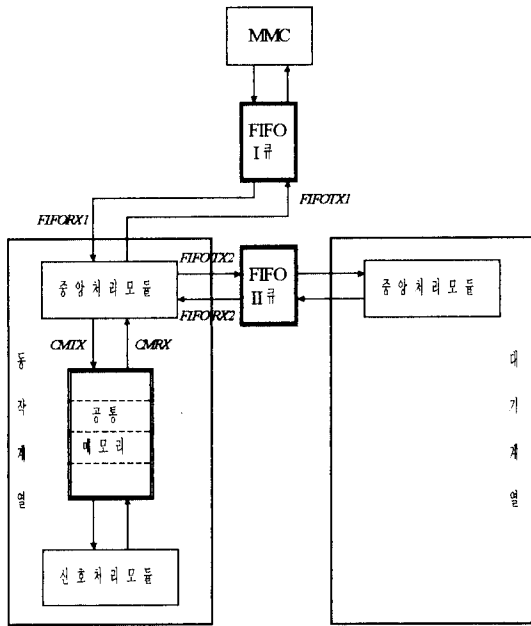


그림 3. 이중화된 ISDN 사설교환기의 인터페이스 구조
Fig. 3. Interface structure of Duplicated ISDN PABX.

4. 이중화 관련 TASK 및 모듈의 상호관계

이중화 TASK가 계열변경과 복구를 하여 시스템 이상시에도 서비스가 계속되도록 하기 위해선 시스템내의 여러 TASK 및 모듈과의 상호관계가 요구되며 이들간의 상호관계와 동작은 그림 4와 같다. 동작계열과 대기계열로 이중화된 교환시스템에서 동작계열은 평상시에 각종 제어정보를 대기계열과 교환하여 계열상호간에 동일상태를 유지한다. 다시 말해 호의 접속이나

단절시 동작계열의 음성 혹은 데이터 호처리 TASK가 제어정보를 대기계열로 보내어 대기계열에서 호 제어 정보의 유지와 호의 접속 및 단절작업을 할 수 있게 한다. 그리고 동작계열의 시스템 감시TASK는 대기계열의 상태를 주기적으로 점검하여 이에 대한 정보를 기록해 두고, 만약 대기계열에 이상이 있으면 시스템 구성TASK로 하여금 리로딩이나 리스타트를 하게 하여 대기계열의 정상 동작을 유지시킨다.

만약 동작계열의 수행중에 치명적인 하드웨어 오류나 소프트웨어 오류가 발생하면 시스템 진단TASK나 하드웨어 자체의 오류검출 회로가 이를 검출하고 대기계열로 난마스크블(nonmaskable) 계열변경 인터럽트를 발생시킨다. 혹은 시스템 운용자가 MMC를 이용하여 대기계열로 난마스크블 계열변경 인터럽트를 발생시킬 수도 있다. 계열변경 인터럽트가 발생하면 처리루틴은 이중화 TASK로 계열변경 명령을 내리고 제어를 이중화 TASK로 넘긴다. 그러면 이중화 TASK가 대기계열이 새로운 동작계열로 되살아나게 하는 계열변경작업을 수행하여 진행중인 서비스가 계속되게 한다. 계열변경이 정상적으로 이루어지고 난 후 새로운 동작계열은 문제가 된 동작계열로 리로딩이나 리스타트 명령을 내려 복구하여 대기계열로서 동작하도록 만든다.

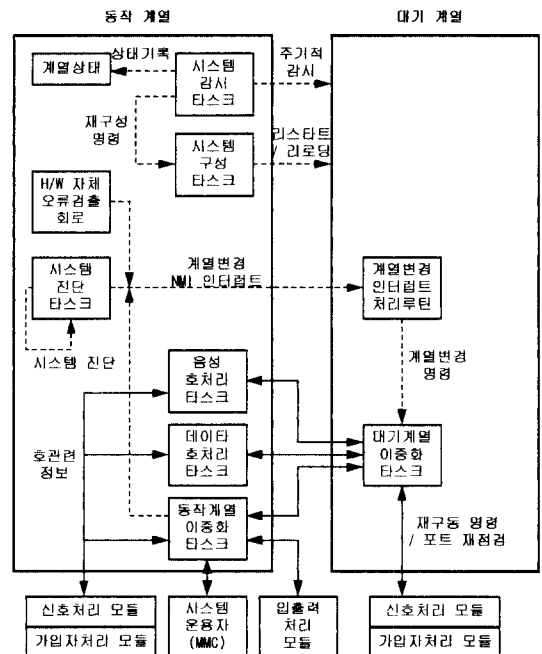


그림 4. 이중화 관련 TASK 및 모듈의 상호관계
Fig. 4. Relationship of tasks and modules related to duplicated system.

IV. 결합내성의 구현방법

교환시스템의 이중화 구현을 위해 이중화 TASK의 수행영역에 대한 결정방법, 시스템 이상시 이중화 TASK가 최우선 순위를 갖도록 하는 방법, 계열변경으로 인한 호제어 정보의 소실을 최소화하기 위한 방법, 불가피한 호제어 정보의 소실로 인한 문제를 해결하는 방법, 계열변경으로 인한 음성이나 데이터 자체의 소실 문제를 해결하기 위한 방법을 제안하였다.

1. 이중화 TASK의 수행영역 결정방법

시스템상에 동작하는 TASK(혹은 프로세스)는 그 수행영역에 따라 시스템(혹은 커널) 영역에서 동작하는 시스템 TASK와 사용자 영역에서 동작하는 사용자 TASK로 나뉜다. 이중화의 기능 구현을 위해선 이를 시스템 영역내 인터럽트 서비스루틴이나 시스템 TASK로 구현할 것인지, 아니면 사용자 영역내 사용자 TASK로 구현할 것인지를 우선 결정하여야 한다.

시스템 이상시 계열변경과 복구는 즉시 행해져야 하는 긴급한 동작이므로 이중화 기능을 시스템 영역내 인터럽트 서비스루틴에 구현할 경우 난마스크별 인터럽트 서비스루틴에 구현되어야 하고, 이중화 기능의 수행을 위해선 소프트웨어 트랩을 이용한 운영체제의 시스템호출이 불가피하다. 그런데 난마스크별 인터럽트의 특성상 이에 비해 우선순위가 낮은 소프트웨어 트랩은 블로킹(blocking)되므로 운영체제의 시스템호출이 불가능하게 된다. 즉, 이중화 기능을 인터럽트 서비스 루틴내에 구현하면 인터럽트의 우선순위 문제가 발생하므로 이는 불가하다.

이중화 기능을 시스템 영역내 시스템 TASK로 구현할 경우 운영체제의 구조상 계열변경 인터럽트의 발생시 시스템 이벤트 리스트에 이중화 TASK를 등록하고 이는 10 msec 타이머 인터럽트의 발생시나 운영체제 시스템호출의 처리 후 디스페치 루틴에 의해 구동되게 된다. 이 또한 이중화 TASK를 구동하는 디스페치 루틴까지는 시간 지연이 있게 되므로 실행의 긴급성에 위배된다. 그러므로 이중화 기능은 사용자 영역내 사용자 TASK로 구현되어야 한다.

2. 이중화TASK의 최우선화 방법

평상시 이중화 TASK는 다른 TASK에 비해 상대적으로 우선순위가 낮다. 이로 인해 계열변경시 우선순위가 높은 다른 TASK에게 계속적으로 제어를 빼앗겨

긴급한 이중화 TASK는 수행하지 못하는 문제(starvation problem)가 발생하게 된다. 그러므로 시스템 이상시에는 이중화 TASK가 최우선 순위(top priority)를 갖도록 하여야 한다. 또한 이중화 TASK는 시스템에 발생한 치명적인 오류를 치유하는 긴급한 작업을 수행해야 하므로 계열변경과 복구작업을 수행하는 동안에는 최우선 순위를 계속 유지하여 다른 TASK에 의해 제어를 빼앗기지 않도록 해야 한다. 만약 이중화 TASK가 계열변경과 복구작업을 수행하는 동안에 최우선 순위를 유지하지 못하면 TASK의 우선순위 문제로 인해 예기치 못한 여러 문제가 발생 가능하다.

계열변경 인터럽트는 난마스크별이므로 서비스 루틴 내에서는 운영체제의 시스템호출이 불가능하여 직접 이중화 TASK의 우선순위를 변경할 수가 없다. 그래서 이중화 TASK로 긴급(emergency) 메시지를 전송함으로써 이중화 TASK가 최우선 순위를 갖도록 만든다. 그런데 긴급메세지 기능은 메세지 전송시에는 해당 TASK가 최상위 우선순위를 갖도록 하여 그 메세지를 빨리 수신할 수 있게 하지만 메세지의 수신과 동시에 TASK 자신의 본래 우선순위로 되돌리기 때문에 또 다시 제어를 빼앗길 위험이 내포되어 있다. 그래서 이중화 TASK는 긴급메세지를 수신하면 곧 바로 시스템 호출을 이용하여 자신의 우선순위를 변경함으로써 최상위 우선순위를 계속 유지할 수 있게 한다. 그리고 이중화 TASK가 계열변경과 복구작업을 성공적으로 종료하여 대기계열이 새로운 동작계열로서 동작할 수 있는 상태가 되었을 때 본래 자신의 우선순위로 되돌린다.

3. 호제어정보 소실의 최소화 방법

동작계열의 음성 또는 데이터 호처리 TASK는 호의 접속이나 단절시 제어정보를 대기계열로 보내어 대기계열에서 호제어 정보의 유지와 호의 접속 및 단절 작업을 할 수 있도록 한다. 그러나 계열변경으로 인하여 계열간 동일상태 유지를 위한 제어정보를 잃어 버릴 수 있으므로 이를 최소화하기 위한 방법이 요구된다. 이를 최소화하기 위해 이중화 TASK는 계열변경 인터럽트 서비스루틴이 긴급메세지 큐를 통해 전송한 계열변경 요구메세지를 자신의 일상메세지 큐로 재 전송한다. 이리하여 동작계열이 FIFO2 큐에 저장한 호제어 정보까지는 계열변경이 발생하더라도 살릴 수 있게 된다.

이의 상세 설명을 위해 동작계열이 호제어 정보를

FIFO2 큐에 저장하고 FIFOTX2 인터럽트를 대기계열로 발생시킨 시점에 시스템에 이상이 생겨 너마스쿠벌 계열변경 인터럽트가 발생하는 최악의 상황을 가정해 보자. FIFORX2 인터럽트 처리루틴은 계열변경 너마스쿠벌 인터럽트에 의해 수행이 일시 중단되고, 계열변경 인터럽트 처리루틴은 이중화 타스크를 최우선 타스크로 만들기 위해 이중화 타스크의 긴급메세지 큐로 계열변경 요구메세지를 전송한다. 그런후 중단된 FIFORX2 인터럽트 처리루틴의 수행이 재개되어 FIFO2 큐로부터 호제어 정보를 읽어 FIFORX2 버퍼에 저장하고, FIFORX2 이벤트를 시스템 이벤트 리스트에 등록한다. 그러면 운영체제의 디스페치 루틴이 FIFORX2 버퍼에서 이중화 타스크의 일상 메세지 큐로 호제어 정보를 전송하는 FIFORX2 이벤트 처리를 한 후, 타스크 스케줄링에 의해 최우선 순위인 이중화 타스크가 구동된다. 이중화 타스크는 자신의 긴급메세지 큐로부터 계열변경 요구메세지를 읽게되지만 계열변경 작업을 즉시 수행하면 이전에 전달된 일반메세지 큐내의 호제어 정보를 잃어버리게 되므로 계열변경 작업의 수행을 일시적으로 유보한다. 즉, 계열변경 요구메세지를 자신의 일상 메세지 큐로 재 전송하여 앞서 전달된 호제어 정보를 처리한 후, 계열변경과 복구작업을 수행하도록 한다.

4. 호제어정보 소실문제의 해결방법

제안한 최소화 방법으로 계열변경으로 인한 호제어 정보의 소실을 극소화하였지만 불가피한 소실이 있을 수 있다. 계열간 동일상태를 유지하기 위해 동작계열은 대기계열로 FIFO2 큐를 통해 호의 접속 및 단락 제어 정보를 보낸다. 하지만 제어정보를 전달하는 과정에서 계열변경이 일어나면 FIFOTX2 메세지 버퍼에 저장된 호제어 정보 - FIFO2 큐에 저장된 호제어 정보를 대기계열에서 읽는 중에 발생한 호제어 정보는 일시적으로 FIFOTX2 메세지 버퍼에 저장됨 - 는 잃어버리게 되므로 그로 인해 발생가능한 문제의 해결이 요구된다.

계열변경의 수행중 성립호 제어정보를 소실하게 되면 계열변경후 교환기는 사용자의 새로운 통화 요청으로 인식하여 다이얼 톤을 내려보낸다. 그러나 교환기의 계열변경은 사용자가 느끼지 못할 정도의 극히 짧은 시간에 이루어지므로 사용자에게 상대방 호출음이 잠시 들리다 혹은 극히 짧은 시간동안 통화연결이 되었다가 다시 다이얼 톤이 들려와도 이는 크게 문제될 것이 없

다. 하지만 단락호 제어정보일 경우는 좀 더 문제가 심각하다. 사용자가 통화를 종료하는 순간에 계열변경이 일어나서 단락호 제어정보를 잃게 되면 대기계열에서는 계속 통화 중인 것으로 남아 있으므로 엄청난 과금 문제가 발생한다. 그래서 계열변경후 통화중 상태의 모든 가입자 포트로 응답요구(Answer Back Request: ABR) 메세지를 내려보내 계열변경중에 단락된 호가 없었는지를 직접 재 점검함으로써 단락호 제어정보의 소실문제를 해결한다.

5. 음성과 데이터 소실문제의 해결방법

계열변경으로 소실가능한 정보는 제어정보일 수도 있지만 음성이나 데이터 자체일 수도 있다. 그러므로 실시간으로 진행중인 음성이나 데이터가 계열변경으로 소실되더라도 사용자가 이를 전혀 느낄수 없도록 하여야 한다. 계열변경으로 인한 음성 자체의 소실은 계열변경의 수행시간과 밀접한 관계가 있고, 데이터의 경우는 데이터통신 프로토콜과 밀접한 관계가 있다. 데이터인 경우에는 패킷 단위로 송수신이 이루어지고 통신중선로상의 문제나 교환기의 계열변경으로 패킷이 소실되더라도 송수신 단말의 엔드-엔드(end-to-end) 통신 프로토콜에 의해 재전송이 이루어지기 때문에 크게 문제되지 않지만 음성인 경우에는 다르다. 음성은 엄격한 실시간 제약이 따르므로 소실된 음성이 50 msec를 넘어서면 잡음으로 들리고 최악의 경우 음절이나 단어의 소실이 가능하므로 수화자의 청취가 곤란할 수 있다^[13]. 그러므로 계열변경의 수행시간이 50 msec 이내가 되도록 하여야 한다.

V. 결합내성의 구현

계열변경과 복구기능은 이중화의 핵심 기능으로써 소프트웨어 혹은 하드웨어 오류가 발생하거나 시스템 운용자의 필요에 의한 계열변경 요구시에 계열변경과 복구를 수행하여 대기계열이 새로운 동작계열로서 사용자에게 서비스를 계속 제공할 수 있도록 하는 기능을 수행한다. 이중화 기능의 핵심이 되는 계열변경과 복구를 제안한 구현방법에 의거하여 구현하였으며 그림 5에 이의 데이터 흐름을 나타내었다.

- (1) 동작계열의 시스템 오류나 시스템 운용자의 필요에 의해 계열변경이 요구되면 대기계열로 너마스쿠벌 계열변경 인터럽트를 발생시킨다.

- (2) 계열변경 인터럽트 서비스 루틴은 이중화 타스크를 최우선 순위 타스크로 만들기 위해 계열변경 요구메세지를 만들어 이를 이중화 타스크의 긴급메세지 큐로 송신한다.
- (3) 이중화 타스크가 긴급메세지 큐로부터 계열변경 요구메세지를 수신하면 우선 자신의 우선순위를 최고로 유지하기 위해 우선순위 변경을 한 후, 계열변경 시작 시각을 기록하고, 현재 계열이 동작계열로 바뀌었다는 계열변경 메세지를 신호처리모듈로 송신하여 이중화타스크가 동작하기 위한 최소한의 환경을 구축한다.
- (4) 동일상태 유지를 위해 전달된 각종 제어메세지들이 운영체제의 디스패치 루틴에 의해 이중화 타스크의 일상 메세지 큐로 옮겨진다.
- (5) 전달된 각종 이중화 관련 메세지를 우선적으로 처리한 후에 계열변경 작업을 수행하기 위하여 계열변경 요구메세지를 이중화 타스크 자신의 일상 메세지 큐로 재 전송한다.
- (6) 이중화 타스크는 자신의 일상메세지 큐로부터 메세지를 읽어 각종 제어메세지를 우선 처리한 후, 계열변경 요구메세지를 읽게 되면 계열변경 작업을 시작한다.
- (7) 계열변경을 위해 가입자처리 모듈로 재구동 시작 메세지를 송신하고, 계열변경의 수행중에 단락된 호가 없었는지를 확인하기 위해 가입자처리 모듈내의 현재 통화중 상태인 모든 가입자 포트에 응답요구 메세지를 내려 보낸후, 가입자처리 모듈로 재구동 종료 메세지를 송신한다. 이때 실행된 모든 가입자처리 모듈에 대해 이 과정을 반복 수행한다.
- (8) 계열변경 결과를 출력하기 위해 시스템 운용자의 모니터, 입출력처리 모듈 및 하드디스크로 계열변경 결과메세지를 전송한다.
- (9) 계열변경을 성공적으로 완료하여 새로운 동작계열로 살아나 가입자의 요구에 따른 서비스를 계속한다. 계열변경이 정상적으로 이루어지고 난 뒤 새로운 동작계열은 문제가 된 동작계열로 리로딩이나 리스타트 명령을 내려 이를 복구하여 대기계열로서 동작하도록 한다. 만약 이상이 발생한 계열을 가능한 빨리 복구하기 위하여 계열변경의 수행 초기에 리로딩이나 리스타트 명령을 내렸는데 불행히도 계열변경의 수행중 또 다른 이상으로 문제가 된 계열로 계열변경을 요청을 하게 되면 양측계열 모두 다 운이 되게 된다. 그러므로 문제가 된 계열의 복구는 계열변경의 마지막 단계에서 행하여야 한다.
- (10) 계열변경 완료시각을 기록하고, 이중화 타스크의 우선순위를 본래 이중화 타스크 자신의 우선순위로 되돌린다.

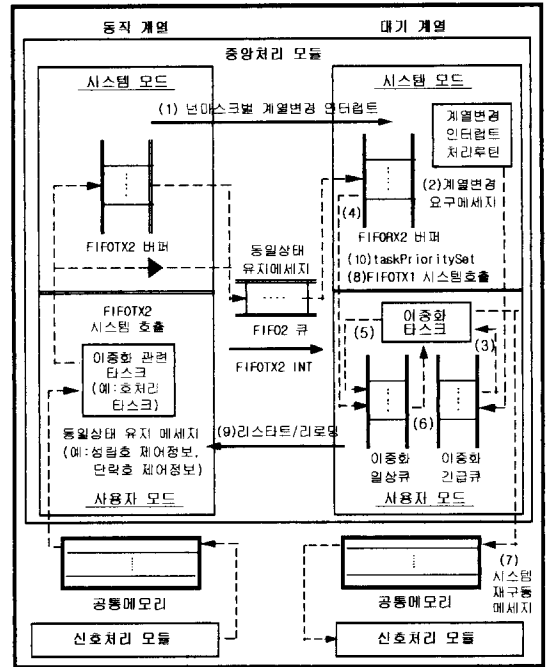


그림 5. 계열변경과 복구의 데이터 흐름도

Fig. 5. Dataflow diagram of side exchange and recovery.

VI. 성능평가

본 연구의 궁극적인 목표는 시스템에 어떤 이상이 발생하더라도 사용자측에선 이러한 결함을 전혀 느끼지 못한 상태에서 진행중이던 음성 혹은 데이터 통신서비스를 계속해서 제공받을 수 있도록 하는데 있다. 이를 위해선 계열변경으로 인해 발생하는 호제어 정보와 음성 혹은 데이터 자체의 불가피한 소실 문제를 해결하여야 한다. 호제어 정보의 소실 문제는 앞서 결합내성의 구현방법에서 상세히 설명하였듯이 사용자에게 다 이일 돈을 내려보내거나 통화중 상태의 가입자 포트를 직접 재 점검함으로써 해결이 가능하다. 하지만 계열변경으로 인한 음성이나 데이터 자체의 소실은 계열변경의 수행시간과 밀접한 관계가 있으므로 이에 관해 성능을 평가하였다.

1. 성능평가 방법

시스템 이상시 대기계열이 계열변경과 복구를 완료하고 새로운 동작계열로 되살아나기까지 소요되는 총 수행시간을 측정함으로써 이중화의 성능평가를 하였다. 이를 위해 우선 가변적인 시스템 변수는 일반적인 상

수값으로 대치하고 발생할 가능성이 희박한 상황은 무시하여 타당한 전제조건을 설정하였다. 다음으로 계열 변경과 복구를 위한 이중화의 수행흐름을 분석하여 수행단계를 파악한 후, 설정한 전제조건하에 시스템 통신 트래픽의 강도를 달리하여 각 단계의 최단, 일상 및 최장 수행시간을 모토롤라 68000 계열의 디버깅 툴인 *XRAY68K*^[9]를 이용하여 측정하였다. 그리고 각 단계별 수행시간이 구해지면 이의 수행횟수를 곱하여 단계별 총 수행시간을 구하고, 이를 합산하여 계열변경과 복구에 소요되는 총 수행시간을 구하였다.

*XRAY68K*를 이용하여 측정할 수행시간의 기본단위는 클럭이며 본 ISDN 교환기의 중앙처리모듈은 모토롤라 68020 마이크로프로세서를 사용하여 CPU 속도는 16 MHz이므로 한 클럭 사이클의 시간은 62.5×10^{-9} ($=1/(16 \times 10^6)$) 초이다. 그래서 *XRAY68K*를 이용하여 측정할 총 클럭수에 62.5×10^{-9} 를 곱하여 시간으로 환산하였다. 그리고 각 단계의 수행시 요구되는 운영체제와 데이터베이스의 수행시간은 참고문헌^[10]의 일상 수행시간을 참조하였다.

2. 성능평가의 전제조건

계열변경과 복구에 소요되는 총 수행시간을 구하기 위해 먼저 가변적인 시스템 변수는 일반적인 상수값으로 대치하고, 발생 가능성이 희박한 상황은 무시하여 다음과 같은 전제조건을 설정하였다^[4,10].

- (1) ISDN 사설교환기는 단일 노드로 구성되어 있다.
- (2) 8 개의 가입자 처리모듈이 노드내에 모두 정상 실장되어 있다.
- (3) 각 가입자 처리모듈의 가입자 수용 용량은 256 회선이다.
- (3) 계열간 동일상태 유지를 위한 메시지의 평균 길이는 170 바이트이다.
- (4) 계열변경을 위한 재구동메세지의 평균 길이는 10 바이트이다.
- (5) 이중화 TASK의 일상메세지큐의 크기는 1024 바이트이다.
- (6) FIFO2 메세지 버퍼의 크기는 1024 바이트이다.
- (7) FIFO2 큐의 크기는 512 바이트이고, 단일 메세지 블록으로 구성되어 있다.
- (8) 공통메모리의 크기는 1024 바이트이고, 각 블록의 크기가 50 바이트인 20 개의 메세지 블록으로 구성되어 있다.
- (9) 계열변경 결과메세지의 출력은 무시한다.
- (10) 시스템의 통신트래픽 강도는 FIFO2 큐, FIFO2 버

퍼 및 이중화 일상큐내에 처리 대기중인 메세지의 수와 통화나 통신중인 가입자의 수에 따라 결정된다.

3. 이중화의 수행흐름

이중화의 수행흐름은 앞서 상세히 설명한 바와 같이 크게 동일상태 유지를 위한 수행과 계열변경과 복구를 위한 수행으로 나눌수 있다. 우선 계열간 동일상태의 유지를 위해 동작계열은 발생한 호의 제어정보를 FIFO2 큐에 쓴 후, FIFOTX2 인터럽트를 대기계열로 발생시킨다. 그러면 대기계열의 FIFORX2 인터럽트 처리루틴이 구동되어 FIFO2 큐에 수신된 호제어 정보를 읽어 FIFORX2 버퍼에 저장해 두면 이를 시스템 TASK가 이중화 TASK의 일상메세지큐로 이동시킨다. 그러면 대기계열의 이중화 TASK가 이를 읽어 요구된 명령에 따라 호의 접속과 단절, 데이터베이스의 변경 작업을 하여 대기계열이 동작계열과 동일상태를 유지하도록 한다. 다시 말해 동일상태 유지를 위한 수행은 동일상태 유지메세지의 전달단계와 동일상태 유지메세지의 처리단계를 거친다. 동작계열이 정상 동작시는 동일상태의 유지를 위한 작업만이 수행되지만 일단 동작 계열에 이상이 생기면 계열변경과 복구를 위한 작업으로 넘어간다. 이는 앞서 계열변경과 복구 절차에서 상세 설명하였듯이 계열변경 인터럽트의 발생으로부터 계열변경 요구메세지가 이중화 일상 메세지 큐에 저장되기까지의 계열변경 요구메세지의 전달단계와 대기계열의 이중화 TASK가 이를 읽어 계열변경과 복구를 수행하는 계열변경 요구메세지의 처리단계를 거친다.

4. 계열변경과 복구의 성능평가

계열변경과 복구에 소요되는 총 수행시간은 시스템의 통신트래픽 강도에 따라 가변적이며, 이는 FIFO2 큐, FIFO2 버퍼 및 이중화 일상큐내에 처리 대기중인 메세지의 수와 통화나 통신중인 가입자의 수에 따라 결정된다. 다시 말해 FIFO2 큐, FIFO2 버퍼 및 이중화 일상큐내에 존재하는 계열간 동일상태 유지를 위해 동작계열로부터 전달된 메세지의 수에 따라 계열변경과 복구의 총 수행시간이 결정된다. 또한 사용자가 통화를 종료하는 순간 계열변경이 일어나 단락호 제어정보를 잃게 될 경우에 발생하는 문제를 해결하기 위해 계열변경후 통화나 통신중인 포트를 직접 재 점검해야하므로 통화나 통신중인 가입자의 수에 따라 계열변경과 복구의 총 수행시간이 결정된다.

1) 계열변경과 복구의 최단시간

계열변경과 복구의 최단시간은 시스템의 통신트래픽 강도가 0 % 일 때의 계열변경과 시스템 복구를 위한 이중화의 총 수행시간을 말한다. 즉, FIFO2 큐, FIFO2 버퍼 및 이중화 일상큐내에 처리해야할 동일상태 유지메세지가 하나도 없고, 계열변경시 통화나 통신중인 가입자가 하나도 없었을 때의 계열변경과 복구를 위한 총 수행시간이다. 처리해야 할 동일상태 유지메세지가 없으므로 동일상태 유지메세지의 전달과 처리는 불필요하고, 단지 계열변경 요구메세지의 전달과 처리만이 요구된다. 계열변경 요구메세지의 처리시 실장된 모든 가입자 처리모듈로 재구동 시작메세지와 종료메세지를 내려보내야 하므로 16 회의 CMTX 시스템호출이 요구된다. 그러므로 각 단계별 수행시간, 수행횟수 및 총 수행시간으로 구성된 이중화의 수행시간은 표 1과 같고, 계열변경과 복구를 위한 최단 수행시간은 11.8 msec 이다.

표 1. 계열변경과 복구를 위한 최단 수행시간 (조건: 트래픽 강도 = 0 %)

Table 1. Minimum execution time for side exchange and recovery. (단위: μsec)

조건	대기중인 메세지의 수 = 0, 통화중인 가입자의 수 = 0				
	구분	수행시간	수행횟수	총수행시간	
계열변경 요구메세지 전달단계	계열변경 INT 서비스루틴 (절체ISR → 이중화긴급큐)	149.9	1	149.9	
	계열변경 환경구축 루틴	246.6	1	246.6	
	계열변경 요구 전달루틴 (이중화긴급큐 → 이중화큐)	253.8	1	253.8	
	시스템 호출	taskPrioritySet 루틴	49.3	1	49.3
		dbReadAll 루틴	125.0	1	125.0
		CMTX 루틴	73.9	1	73.9
msgQSend 루틴		138.1	1	138.1	
계열변경 요구메세지 처리단계	계열변경 요구 처리루틴 (계열변경과 시스템복구)	9211.3	1	9211.3	
	시스템 호출	msgQReceive 루틴	115.6	1	115.6
		CMTX 루틴	73.9	16	1182.4
		FIFOTX1 루틴	84.6	2	169.2
		FIFOTX2 루틴	84.6	1	84.6
		taskPrioritySet 루틴	49.3	1	49.3
				11849.0	

2) 계열변경과 복구의 일상시간

계열변경과 복구의 일상시간은 시스템의 통신트래픽

강도가 50 % 일 때의 계열변경과 복구를 위한 이중화의 총 수행시간을 말한다. 즉, FIFO2 큐, FIFO2 버퍼 및 이중화 일상큐내에 동일상태 유지메세지가 50 % 정도 차있고, 계열변경시 전체 가입자의 50 % 정도가 통화나 통신중일 때의 계열변경과 복구를 위한 총 수행시간이다. 통신트래픽 강도가 50 % 이면 FIFO2 큐에 1 개, FIFO2 버퍼에 3 개, 이중화 일상큐에 3 개의 동일상태 유지메세지가 있게 되고, 각 가입자 처리모듈의 256 가입자중 128 가입자가 통화나 통신중 상태에 있게 되고, 한 노드내에는 8 개의 가입자 처리모듈이 있으므로 계열변경시 통화나 통신중인 총 가입자 수는 1024 가입자가 된다.

시스템 타스크는 FIFORX2 버퍼에 있는 모든 메세지를 한번에 이중화 일상큐로 이동시키므로 동일상태 유지메세지의 전달단계는 한번만 수행하지만 처리단계에선 하나씩 차례로 모든 메세지를 처리해야 하므로 7 회 수행하게 된다. 그리고 처리단계에서 요구되는 CMTX 시스템호출의 수는 다음과 같다. 우선 실장된 모든 가입자 처리모듈로 재구동 시작과 종료메세지를 내려보내기 위해 16 회, 공통메모리 통신은 동시에 최고 20 개의 메세지를 동시에 전송 가능하므로 한 가입자 처리모듈내의 통화나 통신중인 128 가입자 포트로 재응답메세지를 내려보내기 위해 7 (=128/20)회, 그리고 노드내에는 8 개의 가입자 처리모듈이 있으므로 총 72 (=7×8+16)회의 CMTX 시스템호출이 요구된다. 그러므로 각 단계별 수행시간, 수행횟수 및 총 수행시간으로 구성된 이중화의 수행시간은 표 2와 같고, 계열변경과 복구를 위한 일상 수행시간은 20.1 msec 이다.

3) 계열변경과 복구의 최장시간

계열변경과 복구의 최장시간은 시스템의 통신트래픽 강도가 100 % 일 때의 계열변경과 시스템 복구를 위한 이중화의 총 수행시간을 말한다. 즉, FIFO2 큐, FIFO2 버퍼 및 이중화 일상큐가 수용할수 있는 최대한으로 동일상태 유지메세지가 차있고, 계열변경시 모든 가입자가 통화나 통신중일 때의 계열변경과 복구를 위한 총 수행시간이다. 통신트래픽 강도가 100 % 이면 FIFO2 큐에 3 개, FIFO2 버퍼에 6 개, 이중화 일상큐에 6 개의 동일상태 유지메세지가 있게 되고, 각 가입자 처리모듈당 256 가입자가 있고 한 노드내에는 8 개의 가입자 처리모듈이 있으므로 계열변경시 통화중인 총 가입자 수는 2048 가입자가 된다.

시스템 타스크는 FIFORX2 버퍼에 있는 모든 메세

지를 한번에 이중화 일상큐로 이동시키므로 동일상태 유지메세지의 전달단계는 한번만 수행하지만 처리단계에선 하나씩 차례로 모든 메세지를 처리해야 하므로 15 회 수행하게 된다. 그리고 처리단계에서 요구되는 CMTX 시스템호출의 수는 다음과 같다. 우선 실장된 모든 가입자 처리모듈로 재구동 시작과 종료메세지를 내려보내기 위해 16 회, 공통메모리 통신은 동시에 최고 20 개의 메세지를 동시에 전송 가능하므로 한 가입자 처리모듈내의 통화중인 256 가입자 포트에 대응된 메세지를 내려보내기 위해 13 (=256/20)회, 그리고 노드내에는 8 개의 가입자 처리모듈이 있으므로 총 120 (=13×8+16)회의 CMTX 시스템호출이 요구된다. 그러므로 각 단계별 수행시간, 수행횟수 및 총 수행시간으로 구성된 이중화의 수행시간은 표 3과 같고, 계열변경과 복구를 위한 최장 수행시간은 27.7 msec 이다.

표 2. 계열변경과 복구를 위한 일상 수행시간 (조건: 트래픽 강도 = 50 %)

Table 2. Average execution time for side exchange and recovery.

(단위: μsec)

조 건	대기중인 메세지의 수=7, 통화중인 가입자의 수=1024				
단 계	루틴	수행시간	수행횟수	총수행시간	
동일상태 유지메세지 전달단계	FIFORX2 INT 서비스루틴 (FIFO2큐 → FIFORX2버퍼)	108.7	1	108.7	
	시스템 태스크 처리루틴 (FIFORX2버퍼 → 이중화큐)	386.3	1	386.3	
동일상태 유지메세지 처리단계	동일상태 유지 처리루틴 (계열간 동일상태 유지)	255.9	7	1791.3	
	시스템 호출	msgQReceive 루틴	115.6	7	809.2
		libMemcpy 루틴	30.4	7	212.8
		dbRead 루틴	112.5	7	787.5
계열변경 요구메세지 전달단계	계열변경 INT 서비스루틴 (절체ISR → 이중화긴급큐)	149.9	1	149.9	
	계열변경 환경구축 루틴	246.6	1	246.6	
	계열변경 요구 전달루틴 (이중화긴급큐 → 이중화큐)	253.8	1	253.8	
	시스템 호출	taskPrioritySet 루틴	49.3	1	49.3
		dbReadAll 루틴	125.0	1	125.0
		CMTX 루틴	73.9	1	73.9
		msgQSend 루틴	138.1	1	138.1
계열변경 요구메세지 처리단계	계열변경 요구 처리루틴 (계열변경과 시스템복구)	9211.3	1	9211.3	
	시스템 호출	msgQReceive 루틴	115.6	1	115.6
		CMTX 루틴	73.9	72	5320.8
		FIFOTX1 루틴	84.6	2	169.2
		FIFOTX2 루틴	84.6	1	84.6
		taskPrioritySet 루틴	49.3	1	49.3
				20083.2	

표 3. 계열변경과 복구를 위한 최장 수행시간 (조건: 트래픽 강도 = 100 %)

Table 3. Maximum execution time for side exchange and recovery

(단위: μsec)

조 건	대기중인 메세지의 수=15, 통화중인 가입자의 수=2048				
단 계	루틴	수행시간	수행횟수	총수행시간	
동일상태 유지메세지 전달단계	FIFORX2 INT 서비스루틴 (FIFO2큐 → FIFORX2버퍼)	108.7	1	108.7	
	시스템 태스크 처리루틴 (FIFORX2버퍼 → 이중화큐)	386.3	1	386.3	
동일상태 유지메세지 처리단계	동일상태 유지 처리루틴 (계열간 동일상태 유지)	255.9	15	3838.5	
	시스템 호출	msgQReceive 루틴	115.6	15	1734.0
		libMemcpy 루틴	30.4	15	456.0
		dbRead 루틴	112.5	15	1687.5
계열변경 요구메세지 전달단계	계열변경 INT 서비스루틴 (절체ISR → 이중화긴급큐)	149.9	1	149.9	
	계열변경 환경구축 루틴	246.6	1	246.6	
	계열변경 요구 전달루틴 (이중화긴급큐 → 이중화큐)	253.8	1	253.8	
	시스템 호출	taskPrioritySet 루틴	49.3	1	49.3
		dbReadAll 루틴	125.0	1	125.0
		CMTX 루틴	73.9	1	73.9
		msgQSend 루틴	138.1	1	138.1
계열변경 요구메세지 처리단계	계열변경 요구 처리루틴 (계열변경과 시스템 복구)	9211.3	1	9211.3	
	시스템 호출	msgQReceive 루틴	115.6	1	115.6
		CMTX 루틴	73.9	120	8868.0
		FIFOTX1 루틴	84.6	2	169.2
		FIFOTX2 루틴	84.6	1	84.6
		taskPrioritySet 루틴	49.3	1	49.3
				27745.6	

5. 고 찰

음성 혹은 데이터 자체가 계열변경으로 인해 소실되더라도 사용자측에선 전혀 문제되지 않는지를 앞의 성능평가를 바탕으로 검증하고자 한다. 먼저 음성통신의 특성을 살펴보면 음성통신의 경우 통화시 음성과 침묵이 반복되므로 실제 음성이 전달되는 시간은 전체의 40 ~ 50 %에 불과하다¹¹⁾. 그리고 음성 자체의 중복성(redundancy)으로 인하여 음성이 1 ~ 2 % 정도 소실되더라도 수화자의 청취에는 문제가 되지 않는다¹²⁾. 하지만 소실된 음성이 50 msec 범위를 넘어서면 잠음으로 들리고 최악의 경우 음절이나 단어가 소실되어 청취가 곤란할 수 있다¹³⁾. 이러한 음성통신의 특성으로 미루어 볼 때 계열변경과 복구에 소요되는 시간이 11.8 ~ 27.7 msec의 범위내에 있으므로 계열변경중에 음성이 소실이 되더라도 사용자측에는 이를 전

혀 느끼지 못하게 된다.

데이터통신의 경우에는 음성과 같은 실시간 제약성은 없으나 데이터의 소실은 허용되지 않는다^[13]. 그러나 데이터 통신중에 오류가 발생하더라도 통신 프로토콜에 의해 오류가 극복된다^[14]. 예를 들면 ACK나 NACK에 의해 데이터 패킷의 안전한 송수신을 확인하고, 만약 데이터 패킷에 오류가 발생하더라도 CRC 체크에 의해 오류검출이 가능하여 NACK로 응답함으로써 재전송 받게 된다. 또한 타이머 메카니즘의 이용으로 데이터 패킷이 소실되더라도 소실되었음을 알 수 있어 재 전송이 이루어지고, 일련번호를 이용하여 패킷의 중복이나 소실 여부도 알 수 있다. 이러한 데이터통신의 특성으로 미루어 볼 때 계열변경으로 인해 데이터 자체가 소실되더라도 통신 프로토콜에 의해 재전송이 이루어지므로 이 또한 문제되지 않는다.

결과적으로 시스템에 어떤 이상이 발생하더라도 사용자측에선 이러한 결함을 전혀 느끼지 못한 상태에서 진행중이던 음성이나 데이터 통신서비스를 계속해서 제공받을수 있음을 입증하였다.

Ⅶ. 결 론

본 논문에서는 음성과 데이터 통신서비스를 제공하는 ISDN 사설교환기에 예기치 못한 이상이 발생하더라도 진행중이던 가입자 서비스를 계속 유지시킬 수 있는 결합내성을 구현하였다. 결합내성은 ISDN 사설교환기의 어느 특정 하드웨어나 소프트웨어 모듈의 예기치 못한 결함이 전체 시스템 다운을 유발하지 않도록 교환기의 주요 제어부분을 동작계열과 대기계열로 이중화하여 구현하였다. 동작계열은 사용자의 요청에 따른 서비스를 제공하면서 계열변경에 필요한 각종 정보를 수시로 대기계열로 보내어 계열 상호간에 동일상태를 유지할 수 있도록 하고, 대기계열은 동작계열의 오류 발생시에 계열변경을 수행하여 새로운 동작계열로서 서비스를 계속할 수 있도록 하였다. 그리고 제안한 방법에 따라 구현함으로써 계열변경으로 인한 호제어 정보의 소실을 최소화하였고, 불가피한 호제어 정보의 소실로 인한 문제도 성공적으로 해결하였다.

계열변경으로 인한 음성이나 데이터 자체의 소실은 계열변경의 수행시간과 밀접한 관계가 있으므로 이의 성능을 평가하였다. 우선 계열변경과 복구의 수행흐름을 분석하여 수행단계를 파악하고, 설정한 전제조건하

에서 통신트래픽의 강도를 달리하여 각 단계별 최단, 일상 및 최장 수행시간을 측정하였다. 각 단계별 수행시간을 합산하여 계열변경과 복구에 소요되는 총 수행시간을 구한 결과 최단시간은 11.8 msec, 일상시간은 20.1 msec, 최장시간은 27.7 msec 이었다.

음성통신의 경우 실제 음성이 전달되는 시간은 전체의 40 ~ 50 %에 불과하고, 음성 자체의 중복성으로 인하여 음성이 1 ~ 2 % 정도 소실되더라도 문제되지 않는다. 만약 소실된 음성이 50 msec 범위를 넘어서면 최악의 경우 음절이나 단어를 소실할 수 있지만 그렇지 않으면 문제되지 않는다. 이러한 음성통신의 특성으로 미루어 볼때 계열변경과 복구에 소요되는 시간은 11.8 ~ 27.7 msec 범위내에 있으므로 계열변경중에 음성이 소실되더라도 사용자는 이를 전혀 느끼지 못하게 된다. 데이터통신의 경우 음성과 같은 실시간 제약성은 없으나 데이터의 소실은 전혀 허용되지 않는다. 그러나 계열변경으로 인해 데이터 자체가 소실되더라도 통신 프로토콜에 의해 재 전송이 이루어지므로 이 또한 문제되지 않는다.

결과적으로 시스템에 예기치 못한 이상이 발생하더라도 사용자측에선 이러한 결함을 전혀 느끼지 못한 상태에서 진행중이던 음성 혹은 데이터 통신서비스를 계속해서 제공받을 수 있음을 성능평가를 통해 검증하였다.

참 고 문 헌

- [1] P. K. Lala, *Fault-Tolerant and Fault Testable Hardware Design*, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.
- [2] S. I. Jun, J. C. Park, J. M. Kim, J. H. Cho, "Concurrent Realtime Operating System for Telecommunication System", 일본 전자정보통신학회, July 1991
- [3] 전 성익, 박 준철, 조 주현, "TDX-10 제어계 이중화 방식 설계 및 구현", 한국전자통신연구소 전자통신 동향분석, 제 14권, 제 4호, pp.67-79, January 1993
- [4] 삼성전자, *Information Exchange System (INFOREX) 시스템 사용자 매뉴얼*, 통신연구소 교환연구실, July 1993
- [5] 최 재원, 박 인갑, "ISDN 사설교환기의 결합내성 설계 및 구현", 대한전자공학회 추계종합학술대회 논문집, December 1995

- [6] N. H. Vaidya and D. K. Pradhan, "A Fault Tolerance for a System of Duplicated Communicating Processes", *IEEE workshop on Fault-Tolerance Parallel and Distributed Systems*, pp.98-104, 1992.
- [7] L. Soderberg and T. Backstrom, "Man Machine Communication in AXE-10", *GLOBECOM'86*, December 1986.
- [8] Jae-Weon Choe and In-Kap Park etc., "A Study on Fault Tolerance using the Dual Processing in the Real-Time Distributed Control System (DCS)", *JTC-CSCC'94*, Vol. II, pp.978-984, July 1994.
- [9] Applied Microsystems Corporation, *VALIDATE/XEI Source-Level Debugger User's manual for 68000 Family Micro-processors*, 1989.
- [10] 최 재원, 박 인갑, "ISDN 사설교환기용 실시간 운영체제의 구현 및 성능평가", 대한전자공학회 논문지, 제 33권, 제 12호, 1996 (게재예정)
- [11] P. T. Brady, "A Statistical Analysis on-off Patterns in 16 Conversations", *Bell System Technical Journal*, Vol.47, pp.73-91, January 1968.
- [12] J. G. Gruber and N. H. Le, "Performance Requirements for Integrated Voice / Data Networks", *IEEE Journal on selected areas in Communication*, Vol.SAC-1, pp.981-1005, December 1983.
- [13] T. A. Gonsalves and F. A. Tobagi, "Comparative Performance of Voice / Data Local Area Networks", *IEEE Journal on selected areas in Communication*, Vol.7, No.5, pp.657-669, September 1990.
- [14] A. S. Tanenbaum, *Computer Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

저 자 소 개



崔在涇(正會員)

1965年 4月 27日生. 1988年 2月
고려대학교 전산학과(학사). 1990
年 8月 미국 미시간주립대학교 전
산학과(석사). 1995年 8月 전국대
학교 전자공학과(박사). 1990年
10月 ~ 현재 삼성전자 정보통신

연구소 선임연구원. 주관심분야: 운영체제, 네트워크,
정보통신망, 이동통신

朴仁甲(正會員) 第 31卷 A編 第 10號 參照

현재 건국대학교 전자공학과 교수