

# ATM 교환기 과부하제어 연구

正會員 기 장 근\*, 최 진 규\*\*, 김 영 선\*\*\*

## A Study on the Overload Control of the ATM Switching System

Jang-Geun Ki\*, Jin-Kyu Choe\*\*, Young-Sun Kim\*\*\* *Regular Members*

※본 논문은 한국전자통신연구소의 연구비 지원으로 수행되었음.

### 요 약

본 논문은 ATM 교환기 제어계의 과부하제어에 대한 연구로서 과부하시에도 효율적이고 안정성있는 시스템 성능을 유지할수 있도록하는 과부하 제어 알고리즘을 제안하였다. 제안된 제어 알고리즘은 호가 허용될 때마다 1 증가하고, D 시간마다 1씩 감소되는 카운터를 도입하여, 이 카운터 값이 임계치 이상이 될 경우 과부하 상태가 시작되는 것으로 판단하고 과부하 제어 알고리즘이 동작한다. 과부하 제어가 시작되면 프로세서의 점유율을 측정하여 목표치를 넘거나 모자라는 비율대로 D 값을 증감시키게 되며, 과부하 상태에서의 호 수락여부는 카운터 값이 특정값 이하일 경우에만 호를 수락하도록 하여 결국 과부하가 발생하면 허용되는 호의 수를 줄이게 된다. 또한 ATM 교환기의 특성인 셀폭주를 제어하기 위해 셀폭주가 발생한 시간이 오래될수록 새로운 입력 호에 대한 수락 확률을 지속적으로 작게되도록 조절하여 셀폭주를 제어한다. 시뮬레이션 결과, 제안된 알고리즘은 다양한 프로세서 과부하 조건하에서 목표점유율을 유지하며, 셀폭주의 경우에도 셀손실율이 감소함을 보였다.

### ABSTRACT

In this paper, a new overload control scheme is proposed for a control system in ATM switching system. The proposed control scheme includes a counter that counts the number of accepted calls and is decreased at each D time interval. In overload condition of call processor, the control scheme detects overload condition when the

\*공주대학교 전자공학과

\*\*한남대학교 전자공학과

\*\*\*한국전자통신연구소

論文番號:96008-0104

接受日字:1996年 1月 4日

counter value reaches a certain threshold value. Under overload condition, processor utilization is measured and the value of the D is updated according to the difference between measured processor utilization and target utilization. A new call is accepted only if the value of the counter is less than the threshold value. In overflow condition of cell traffic, accept probability of new call is reduced exponentially according to the elapsed time. The results of simulation show that the proposed overload control scheme maintains the target utilization very well under the various processor overload conditions and reduces the cell loss probability under the cell overflow conditions.

## I. 서 론

일반적으로 교환기를 설계할 때 최대 트래픽 처리 용량은 가장 경제적으로 최대의 호를 처리할 수 있도록, 즉 성능과 비용의 비가 최대가 되도록 설계한다. 따라서 입력 트래픽이 계획된 최대처리용량 범위 내에서 입력될 때는 시스템이 무리없이 트래픽을 처리할 수 있으나 트래픽의 폭주가 발생하는 과부하 상태에는 문제가 발생한다. 즉, 교환기의 과부하 상태는 교환기의 최대처리용량을 초과하는 트래픽의 급격한 증가에 따라 시스템 자원의 부족으로 인하여 호처리 지연과 호 설정 실패가 증가하고 따라서 서비스 만족도 및 교환기 성능이 저하되는 상태로서, 이러한 과부하 상태에 대한 적절한 제어가 없을 경우 불완료호 및 재시도호의 증가로 시스템 성능이 점차로 하락되어 시스템이 정지되는 최악의 상황이 발생하기도 한다<sup>[1,2]</sup>. 따라서 분산 구조 형태를 갖는 ATM 교환기에서 과부하 상태를 검출하고 이를 적절히 제어하는 것은 가입자에게 적절한 서비스 만족도(품질)를 제공하고 운영자에게 효율적인 망자원 관리 및 안정된 시스템 운용을 보장하기 위해 필수적으로 연구되어야 할 분야이다.

현재까지 과부하 제어에 관한 연구는 교환기 트래픽 모델링을 통한 과부하 특성 해석, 기존 SPC 교환기(AXE-10, NO.1 A, 5ESS 등)에 적용된 과부하 제어 방식 등 여러 연구 결과들이 발표되었다<sup>[3-12]</sup>. 그러나 본 연구에서 수행하고자 하는 ATM 교환 시스템의 과부하에 관한 연구는 거의 없는 실정으로, ATM 교환시스템은 기존의 교환기와는 달리 입력 호의 종류 및 특성이 다양하며 교환기 제어계 구조 및 입력 호를 처리하는 절차 또한 기존 교환기와는 다르므로 앞으로 개발될 ATM 교환기가 과부하시에도 성능의 저

하없이 안정된 서비스를 제공할 수 있도록 하는 과부하 검출/제어 방안에 대한 연구가 필요하다.

## II. 교환기의 과부하 제어

기존의 과부하 검출 및 제어 알고리즘들은 다음과 같이 크게 4가지로 분류할 수 있으며, 일반적으로 프로세서 점유율과 입력부하를 이용한 방식이 많이 사용되고 있다.

입력 부하에 의한 방식은 과부하시에도 교환기의 프로세서 점유율을 적정 수준으로 유지시킬 수 있도록 교환기에 입력되는 호의 수를 일정한 값 이하로 제한하는 방식으로써, 일정한 값 이상으로 입력되는 호들은 교환기에 의하여 기각된다.

프로세서 점유율에 의한 방식은 프로세서의 점유율을 측정하여 미리 정해진 수준을 초과하거나 감소할 때마다 교환기에 입력되는 호의 수를 특정 값(프로세서 점유율의 함수로 정해짐)으로 제한하는 방식으로, 한국전자통신연구소에서 개발된 TDX-1A에서 사용된 방식이다.

큐 길이에 의한 방식은 프로세서 큐의 길이가 미리 정해진 수준을 초과하거나 감소할 때마다 교환기에 입력되는 호의 수를 특정값(큐길이의 함수)으로 제한하는 방식으로, 스웨덴 Alcatel의 S1240에서 사용되었다.

프로세서 점유율과 입력부하에 의한 방식은 제어 구간과 제어구간의 정수배로 이루어지는 조정구간을 설정하여, 조정구간에서 프로세서 점유율을 측정하여 시스템의 상태를 파악하고, 이를 바탕으로 다음 조정구간의 제어구간동안 입력될 수 있는 호의 수를 결정된 후 교환기의 입력되는 호의 수를 결정된 호의 수에 맞추어 유지될 수 있도록 제어하며, 독일 Siemens의 EWSD에서 사용되었다.

### Ⅲ. ATM 교환기 과부하제어 알고리즘

#### 1. ATM 교환기 구조

본 논문에서 연구된 과부하 제어 알고리즘이 적용될 ATM 교환시스템의 구조는 그림 1과 같으며, 사용자 정보와 제어 정보의 전달을 수행하는 전달망과 이를 제어하는 제어망으로 구성된다<sup>[13]</sup>.

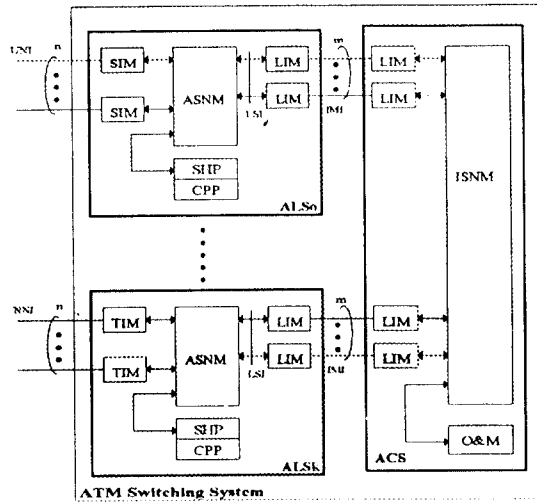


그림 1. ATM 교환기 구조  
Fig. 1 ATM switching system architecture

ATM교환 시스템은 기본적으로 집선 기능을 수행하는 가입자 교환 모듈(ALS)과 ALS 간의 상호접속 기능을 수행하는 중앙 접속 모듈(ACS)로 구성된 모듈화 구조를 가진다. 호처리와 트래픽에 관련된 프로세스들은 ALS내의 CPP내에서 수행되며, 병렬동작의 능력을 수행하는 병렬 프로세서와 성능향상을 위한 직렬 프로세스들로 구성된다.<sup>[13]</sup> ALS는 최소한의 시스템 구성으로서 독립적으로 동작 가능하며, 대규모 시스템에서는 다수의 ALS는 일정한 집선비를 갖는 집선장치로서 동작하고, ACS는 대용량 시스템을 위한 분배장치로서 동작한다. ACS는 ALS들 간의 정보를 교환하는 호처리 과정이 없는 순수한 상호접속기로서 ALS와는 달리 CPP와 SHP를 두지 않는다. ATM 단위스위치는 16×16 단방향 folded 구조로 이루어진 자기루팅 스위치이고 셀들의 내부 및 출력 블러킹 현

상을 해소하기 위하여 공통 메모리를 사용하며, 공통 메모리 내에 저장되어 있는 셀들이 스위칭될 출력 포트를 관리하기 위하여 16개의 어드레스 FIFO버퍼를 두고 있다. 스위치의 작동원리는 매 시간 슬롯 초기에 공통 메모리내에 존재하는 셀 수를 확인한 다음, 버퍼링 규칙에 의하여 입력 포트로부터 셀들을 입력시키고, 출력은 어드레스 버퍼의 최선두에 위치한 셀들을 출력 포트에 내보내어 스위칭이 이루어지게 된다.

#### 2. ATM 교환기 과부하 제어 알고리즘

ATM 교환기는 기존 교환기와는 달리 다양한 입력을 처리하므로 프로세서계의 과부하뿐만 아니라 스위치계의 과부하도 고려할 수 있다. 만일 호수락제어 알고리즘으로 peak bit rate 방식을 사용할 경우 스위치계에서 셀폭주가 발생하지 않을 것이다. 따라서 이 경우 호폭주에 의한 호 과부하 상태만을 고려할 수 있다. 그러나 호수락제어에서 통계적 다중화를 이용한 대역폭 할당 알고리즘 등을 사용한다면 스위치계에서 셀폭주로 인한 과부하가 발생할 수 있다. 따라서 본 논문에서는 먼저 현재 ETRI에서 개발중인 ATM 교환기에서 사용하고 있는 peak bit rate 방식을 가정하여 호폭주시 프로세서 과부하를 제어하는 알고리즘을 제안하고, 그 다음 스위치계에서 통계적 다중화 방식을 사용하는 경우에 발생할 수 있는 셀폭주에 대한 제어를 위해서 공통메모리나 AFIFO 메모리의 셀 수를 검사하여 입력치 이상으로 셀이 있을 경우 셀폭주 발생을 알리는 플래그를 세트하고 이에 따라 호처리 프로세서는 셀폭주 발생시간이 오래될수록 새로운 호에 대한 수락확률을 작게하는 메카니즘을 갖는 알고리즘을 제안하였다.

##### (1) 분산구조 ATM 교환기의 호처리 프로세서 과부하제어 알고리즘

분산구조 ATM 교환기의 제어계에서 호처리 프로세서의 과부하 제어를 위해 본 논문에서 제안한 제어 알고리즘은 근본적으로 입력부하에 의한 방법과 프로세서 점유율에 의한 방법의 조합으로 볼 수 있다. 즉, 본 논문의 과부하 제어 알고리즘은 기존의 방법들과는 달리 허용되는 호의 수를 세는 카운터 C를 두되 이 카운터의 값을 일정 시간간격 D(초기값=평균 호처리시간/목표점유율)마다 감소시킴으로써, 호 요

구가 갑자기 집중되는 경우 카운터 값이 증가하는 구조를 갖는다. 따라서 이 카운터 값이 일정값( $C_{ov-start}$ ) 이상이 될 경우 과부하 상태가 시작되는 것으로 판단하고 과부하 제어 알고리즘이 동작한다. (그림 2 참조) 과부하제어가 시작되면, 일정시간  $W$ 마다 프로세서의 점유율을 측정하여 목표치를 넘거나 모자라는 비율대로 호카운터감소시간간격  $D$  값을 증감시키도록 하였으며, 이는 일반적으로 정확한 평균 호처리시간을 알기가 힘들고 또한 평균호처리시간이 조금씩 변하거나 잘못 판단했을 경우 에도 시스템이 과부하에 대처할 수 있도록 하기 위함이다. 과부하 상태에서의 새로운 호에 대한 수락여부는 카운터 값이 특정 값 이하일 경우에만 호를 수락하도록 하여 결국 과부하가 발생하면 허용되는 호의 수를 줄이게 된다.

다음에 과부하 제어 알고리즘을 교환기의 상태에 따라 과부하 검출, 과부하 제어, 과부하 해제 과정에 대해 상세히 설명한다.

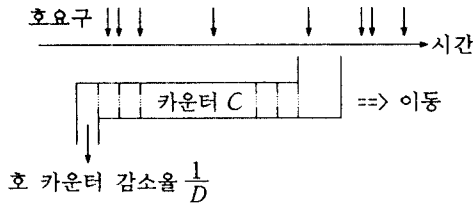


그림 2. 과부하 검출을 위한 카운터의 동작  
Fig. 2 Counter operation for overload detection

① 정상상태

여기서 정상상태라 함은 일반적으로 교환기가 처리할 수 있는 용량 범위내에서 호 요구가 입력되어 요구되는 모든 호를 교환기가 처리할 수 있는 경우를 말한다. 물론 이 경우는 교환기내의 프로세서가 한 호를 처리하는데 걸리는 평균 시간보다 평균 호 요구 시간간격이 당연히 커야 한다.

본 연구에서 적용된 과부하 검출 알고리즘의 동작을 살펴보면 우선 호수 카운터 값은 0으로 초기화되고 호수 카운터 감소 시간간격  $D$ 는 프로세서의 평균 호처리시간  $T_{mean-process}$ 로 초기화 된다. 호가 요구되면 현재 과부하 상태가 시작되지 않았으므로 호를 무조건 수락하고 카운터값을 1 증가 시키게 되며, 카운터

값은 평균 호처리시간마다 1씩 감소된다. 따라서 정상상태의 경우 입력되는 호요구의 랜덤 변화에 따른 정도로만 카운터 값이 변화하고 지속적으로 카운터 값이 증가하는 경우나 큰 값을 유지하는 경우는 발생하지 않는다.

② 과부하 상태 검출

과부하의 시작은 프로세서가 처리할 수 있는 용량 이상으로 호 요구가 증가하는 경우로써, 이 경우 본 연구에서 제안한 과부하 검출/제어 알고리즘의 카운터 값이 계속 증가하게 되며, 임계치  $C_{ov-start}$  이상이 될 경우 과부하제어가 시작된다.

③ 과부하 상태

프로세서가 일단 과부하 상태가 되면 과부하 제어 알고리즘에 따라 과부하 제어가 시작된다. 과부하 제어 알고리즘은 일정시간  $W$ 마다 측정된 프로세서 점유율  $\rho_{real}$ 을 목표점유율  $\rho_{ref}$ 와 비교하여 아래 식에서와 같이 점유율 차이 비율만큼 카운터를 감소시키는 시간간격  $D$ 를 조절한다.

$$D = D \times (1 + (\rho_{real} - \rho_{ref})) \quad \text{if } \rho_{real} > \rho_{ref}$$

$$D = D / (1 + (\rho_{ref} - \rho_{real})) \quad \text{otherwise}$$

또한 과부하제어가 시작되면, 호의 수락여부는 허용 호 갯수 카운터  $C$ 값이 특정값  $C_{threshold}(= C_{ov-start} \times 0.9)$  보다 작을 경우에만 호를 수락하며, 그렇지 않을 경우 거절한다. 위식에서 카운터 감소시간간격  $D$ 는 측정점유율이 목표점유율보다 클 때 증가하며 따라서 카운터 감소가 그만큼 늦게 되기 때문에 호 거절이 상대적으로 늘어나는 효과를 나타내며 결국 프로세서의 부하가 감소하는 결과를 낳는다.

④ 과부하 상태 해제

과부하제어의 해제는 카운터 감소시간간격  $D$ 의 값이 평균호처리시간의 20%에 해당하는 값보다 작아지면 과부하제어를 해제하도록 하였다. 과부하제어를 해제하는 이 값은 실제로 호 수락여부에 크게 영향을 미치지 않는데 그 이유는 현재 과부하 제어 알고리즘이 동작되고 있다하더라도 무조건 호를 거절하는 것이 아니라 카운터 값이 임계치 이상일 경우에

만 호를 거절하도록 되어있고, 실제로 입력 호의 양이 줄어들어 프로세서가 실제로 처리할 수 있는 범위에 들게 되면 카운터 값이 임계치 미만을 유지할 것이기 때문이다.

(2) 사용자 셀폭주를 고려한 과부하제어 알고리즘

앞에서 설명한 과부하 제어 알고리즘은 순수하게 호처리 프로세서의 관점에서만 과부하일 경우에 대한 알고리즘을 기술한 것이다. 만일 수락된 호에서 발생하는 사용자 셀이 스위치의 공통메모리나 AFIFO 메모리의 용량을 초과하여 입력되는 경우 셀손실이 발생할 수 있다. 이 경우에 셀손실을 줄이기 위해 본 연구에서는 셀폭주가 발생하면 수락되는 호의 수를 줄여 셀손실을 줄이도록 하였으며, 이때 셀폭주 제어의 시작과 해제의 판단은 공통메모리나 AFIFO 메모리 중 어느 하나라도 들어있는 셀 수가 각각 임계치  $CM_{threshold\_up}$  또는  $AFIFO_{threshold\_up}$ 를 넘으면 셀폭주 플래그를 세트하여 셀폭주에 대한 제어를 시작하도록 하였고, 공통메모리와 AFIFO 모두가 각각 임계치  $CM_{threshold\_down} (< CM_{threshold\_up})$ 와  $AFIFO_{threshold\_down} (< AFIFO_{threshold\_up})$  보다 작아지는 경우 셀폭주 플래그를 리셋하도록 하였다. 호처리 프로세서는 셀폭주 플래그의 세트 여부에 따라 적절히 새로운 호에 대한 수락확률을 계산하여 호수락여부를 결정한다. 알고리즘의 상세한 내용은 다음과 같다.

먼저 셀폭주를 과부하제어 알고리즘에 반영할 경우 발생할 수 있는 경우는 크게 정상상태, 호처리 프로세서의 과부하만 발생하는 상태, 셀폭주만 발생하는 상태, 호처리 프로세서 과부하와 셀폭주가 모두 발생하는 상태가 있을 수 있다.

정상상태일 경우와 호처리 프로세서만의 과부하일 경우 앞에서 기술한 호처리 프로세서만의 과부하에 대한 제어 알고리즘과 마찬가지로 동작한다. 즉 정상상태일 경우 호처리 프로세서는 요구되는 모든 호를 수락하게 되며, 호과부하만 발생하는 경우는 허용호 수 카운터 값에 따라 호의 수락여부를 결정하게 된다.

셀폭주만 발생할 경우, 즉 새로운 호 요구는 호처리 프로세서의 처리용량 미만으로 입력되나 현재 수락된 호들이 발생하는 셀들이 폭주할 경우 호처리 프로세서는 아래 식과 같이 셀폭주발생시간이 오래 될수록 새로운 호의 수락 확률을 작게 되도록 하여 셀

손실을 줄인다.

$$\text{호수락확률} = a / (\text{현재시간} - \text{셀폭주발생시간} + a)$$

단, a는 상수

호과부하 상태이면서 셀폭주도 동시에 발생할 경우에 호수락여부는 앞에서 설명한 과부하 제어 알고리즘의 허용호 갯수 카운터 C값이 임계치  $C_{threshold}$  보다 클때는 부조건 호를 거절하고 작을 때는 위식과 마찬가지로 셀폭주발생시간이 오래 될수록 호수락확률을 작게한다. 또한 호과부하 상태에서는 측정된 프로세서 점유율과 목표점유율의 차이에 따라 호카운터감소시간간격 D 값을 조절하게 되는데 만일 프로세서 점유율 계산 윈도우 구간 내에서 셀폭주가 있었을 경우에는 허용되는 호수가 증가하지 않도록 하기 위해 호카운터감소시간간격 값을 변경시키지 않는다. 이와 같이 한 이유는 비록 호처리 프로세서의 측정 점유율이 목표점유율보다 낮더라도 셀폭주가 발생할 경우에는 수락호수를 증가시켜서는 안되기 때문에 허용되는 호수가 증가하는 방향, 즉 호카운터감소시간간격 값을 감소시키는 방향으로는 갱신하지 않는 것이다.

그림 3의 (a)에 과부하제어 알고리즘에서 새로운 호에 대한 수락확률을 계산하는 메카니즘을 의사코드

```

/* Cell_OV_Flag : 셀폭주 발생여부 플래그 */
/* Call_OV_Flag : 호폭주 발생여부 플래그 */
/* Pacpt_new_call : 새로운 호에 대한 수락확률 */
/* C : 허용호 갯수 카운터 */
/* Cthreshold : 호과부하시 호수락 카운터 임계치 */
/* Tnow : 현재시간 */
/* Cell_OV_Time : 셀폭주 발생시간 */
/* a : 상수 */

IF Cell_OV_Flag=0 and Call_OV_Flag=0 /* 정상상태 */
THEN Pacpt_new_call = 1

IF Cell_OV_Flag=0 and Call_OV_Flag=1 /* 호과부하 */
THEN IF C<Cthreshold
THEN Pacpt_new_call = 1
ELSE Pacpt_new_call = 0

IF Cell_OV_Flag=1 and Call_OV_Flag=0 /* 셀폭주 */
THEN Pacpt_new_call = a / (Tnow - Cell_OV_Time + a)

IF Cell_OV_Flag=1 and Call_OV_Flag=1 /*셀폭주와 호과부하*/
THEN IF C<Cthreshold
THEN Pacpt_new_call = a / (Tnow-Cell_OV_Time+a)
ELSE Pacpt_new_call = 0
    
```

(a) 호수락확률 계산

```

/*  $\rho_{real}$  : 측정점유율,  $\rho_{ref}$  : 목표점유율 */
/* D : 허용호수카운터감소시간간격 */
/* W : 프로세서 점유율 계산 윈도우 길이*/
IF Call_OV_Flag=1
  THEN IF  $\rho_{real} > \rho_{ref}$ 
    THEN D=Dx(1+( $\rho_{real} - \rho_{ref}$ ))
    ELSE IF no_cell_overflow_in_W
      THEN D=D/(1+( $\rho_{ref} - \rho_{real}$ ))
      ELSE D=D
    ELSE D=D
  
```

(b) 프로세서 점유율에 따른 카운터감소시간간격 계산

그림 3. 호수락확률 및 카운터감소시간간격 계산 알고리즘  
Fig. 3 Algorithms for the probability of call acceptance and the counter decrement interval

(pseudo code) 형태로 나타내었다. 또한 일정시간 W 마다 측정된 프로세서 점유율  $\rho_{real}$ 을 목표 점유율  $\rho_{ref}$  와 비교하여 점유율 차이 비율만큼 카운터감소시간 간격 D를 조절하는 메카니즘을 그림 3의 (b)에 나타내었다.

#### IV. 시뮬레이션 및 결과

본 논문에서 제안된 과부하제어 알고리즘의 성능을 평가하기 위해 앞에서 기술한 ATM 교환기를 토대로 자국호 시나리오를 반영한 시뮬레이션 모델을 SLAM II 언어를 사용하여 개발하고 이를 이용 시뮬레이션을 수행하였다.

3장에서도 언급하였듯이 ATM 교환기의 상태는 크게 정상상태, 호과부하 상태, 셀폭주 상태, 호과부하 및 셀폭주가 동시에 발생하는 상태 등으로 구분할 수 있다. 본 장에서는 개발된 과부하제어 알고리즘의 동작을 검증하기 위해 먼저 정상상태와 호처리 프로세서만의 과부하가 발생하는 상태에 대한 실험 및 검토내용을 기술하였고, 다음에 셀폭주가 발생하는 경우에 대한 결과를 기술하였다.

##### 1. 호처리 프로세서 과부하에 대한 시뮬레이션

시뮬레이션시 포함된 ALS갯수 = 16,  $\rho_{ref} = 0.9$ ,  $W = 5000$ ,  $Cov\_start = 20$ ,  $C_{threshold} = 18$ ,  $T_{mean-process} = 50$ 으로 설정하였으며, 이에 대한 시뮬레이션 결과를 표 1에 요약하였다. 표 1에서 입력부하는 입력호의 양을 프로세서 최대처리능력의 비로 나타낸 것으로 목표점유율 0.9 이상으로 입력이 들어올 경우 과부하제어가

적절히 이루어져 프로세서 점유율이 0.9를 유지함을 알 수 있다.

표 1. 입력부하에 따른 프로세서의 평균 점유율

Table 1. Processor utilization for input load

입력 부하	총시도 호수	거절 호수	수락 호수	호 수락율	평균 큐길이	평균대기시간	평균 점유율
0.7	5608	0	5608	1.0000	2.08	16.52	0.7000
0.8	6382	6	6376	0.9991	3.99	27.84	0.7985
0.9	7183	56	7127	0.9922	9.03	56.36	0.8909
1.0	7996	792	7205	0.9010	9.56	59.05	0.9000
1.5	11954	4751	7203	0.6025	7.66	47.33	0.8998
2.0	15974	8769	7204	0.4510	7.52	46.42	0.8997
2.5	20002	12787	7215	0.3607	7.29	44.96	0.8999
3.0	23966	16757	7209	0.3008	7.41	45.72	0.9002
5.0	39960	32748	7212	0.1805	7.38	45.53	0.9003

또한 시간에 따라 동적으로 변화하는 입력에 대한 결과를 그림 4~6에 나타내었다.

그림 4에는 프로세서 점유율이 10%에서 200%로 지수적으로 증가하였다가 다시 10%로 감소되도록 입력 트래픽 양이 변화할 경우에 대한 결과를 나타내었는데, 그림에서 실선으로 된 부분은 전체 입력 트래픽의 양을 프로세서 점유율로 환산하여 나타낸 것이고 굵은 점으로 표시된 것은 실제 프로세서의 점유율이다. 또한 그림에서 수직선 4개는 왼쪽부터 차례로 입력 트래픽양이 증가하여 프로세서 최대능력의 0.9 이상이 되게 하는 시점, 과부하 검출 시점, 입력 트래픽양이 감소하여 프로세서 최대능력의 0.9 이하가 되게 하는 시점, 과부하 해제 시점을 차례로 나타내며, 그림으로부터 본 연구에서 제안한 과부하 제어 알고리즘은 지수적인 변화를 잘 따라가고 있으며 또한 입

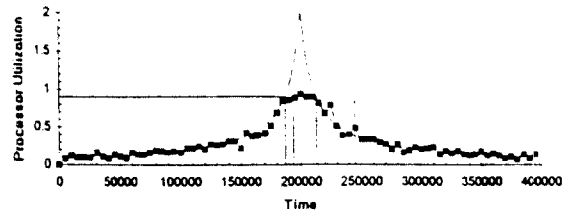


그림 4. 입력 호량 변화에 따른 프로세서 점유율 변화(I)  
Fig. 4 Processor utilization for input load(I)

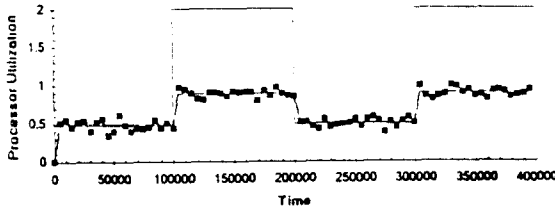


그림 5. 입력 호량 변화에 따른 프로세서 점유율 변화(II)  
Fig. 5 Processor utilization for input load(II)

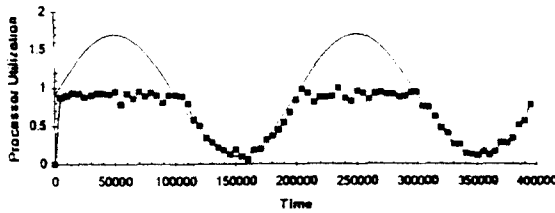


그림 6. 입력 호량 변화에 따른 프로세서 점유율 변화(III)  
Fig. 6 Processor utilization for input load(III)

력 트래픽이 목표점유율 0.9 이상으로 입력될 경우에도 목표점유율 0.9를 잘 유지하며 동작하고 있음을 확인 할 수 있다.

그림 5, 6에는 각각 입력 트래픽 양이 구형과 형태로 갑자기 변화하거나 정현파 형태로 변화할 경우에 대한 결과를 나타내었으며 이 경우 또한 목표점유율 0.9를 잘 유지함을 알 수 있다.

그림 7, 8은 평균 호처리시간을 잘못 판단하거나 판단에 오차가 있는 경우를 가정하여 과부하제어 알고리즘의 파라미터들중 평균호처리시간 값을 전부 20%씩 큰 값 또는 작은 값으로 설정하여 시뮬레이션 한 경우이다. 즉, 이 경우는 과부하 알고리즘이 동작하는 조건 및 파라미터들을 모두 평균호처리시간이 50인 것으로 생각하여 설정하였으나 시스템내의 실제 호처리시간이 가정된 평균호처리시간보다 20% 크거나 또는 20%가 작은 경우에 해당한다. 이 경우들에 대한 결과중 정현파 입력에 대한 예만을 그림 7, 8에 나타내었다. 이 결과로부터 평균 호처리시간을 20% 정도 잘못 판단하여 과부하 제어 알고리즘의 파라미터 값들을 설정하여도 시스템이 발전한다든가 하는 이상 현상은 나타나지 않고 과부하 제어가 비교적 적절하게 이루어 짐을 알 수 있으며, 이 경우는 시

스템의 실제 평균 호처리시간을 아주 정확하게 구하기가 사실상 어렵다는 점을 감안하면 바람직한 결과로 판단된다. 실제 평균 호처리시간이 예상 평균 호처리시간보다 20%가 더 큰 경우에 과부하가 시작되는 초기에 일부 결과에서 점유율이 1에 가까운 구간이 일시적으로 나타났는데 이는 초기 파라미터 설정의 오차에 기인하며, 이러한 현상은 과부하 시작 직후를 프로세서 점유율이 아닌 입력되는 호의 갯수를 가지고 판단하는데서 비롯된다고 분석된다.

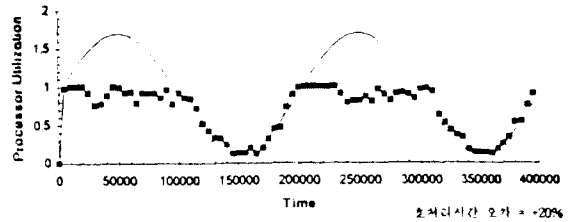


그림 7. 입력 호량 변화에 따른 프로세서 점유율 변화(IV)  
Fig. 7 Processor utilization for input load(IV)

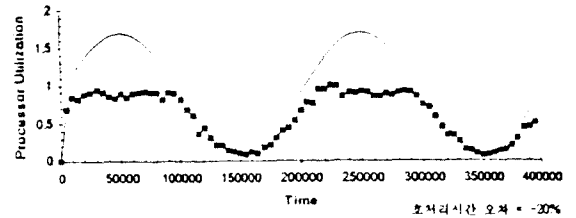


그림 8. 입력 호량 변화에 따른 프로세서 점유율 변화(V)  
Fig. 8 Processor utilization for input load(V)

## 2. 사용자 셀폭주에 대한 시뮬레이션

앞에서 기술한 시뮬레이션 결과들은 사용자 셀폭주가 발생하지 않는다고 가정하고 실험한 결과들이다. 여기서는 ATM 교환기의 스위치에 있는 메모리 버퍼에서 사용자 셀의 폭주가 발생하는 경우에 대한 실험 결과를 기술하였다. 사용자 셀폭주가 발생하는 경우는 크게 호처리 프로세서가 정상상태일때 발생하는 경우와 호처리 프로세서가 과부하 상태이면서 셀폭주도 발생하는 경우가 있다. 이 중 그림 9에 호과부하 상태(호부하량 150%)이고, AFIFO<sub>length</sub> = 256,

AFIFO<sub>threshold\_up</sub> = 128, AFIFO<sub>threshold\_down</sub> = 102일 경우에 대한 실험 결과를 나타내었다. 그림에서 알 수 있듯이 셀폭주를 반영한 알고리즘은 셀입력이 많아짐에 따라 새로운 호에 대한 수락확율을 작게함으로써 수락호수를 줄이고 결과적으로 셀손실율을 줄일 수 있음을 확인할 수 있다. 그림에 나타내진 않았지만 호처리 프로세서는 과부하가 아니면서(호부하량 70%) 셀폭주가 발생하는 경우에도 그림 9와 동일한 결과를 얻었다.

### V. 결 론

본 연구는 ATM 교환기 제어계의 과부하제어에 대한 연구로서 ATM 교환기 구조를 바탕으로, 과부하시에도 효율적이고 안정성있는 시스템 성능을 유지할 수 있도록하는 과부하 검출/제어 알고리즘을 제안하고 시뮬레이션을 통해 성능을 분석하였다.

분산구조 ATM 교환기에서의 과부하 제어를 위해 본 논문에서 제안한 제어 알고리즘은 기존 교환기의 과부하 제어 알고리즘과는 달리 호가 허용될 때마다 1 증가하고 일정시간(호카운터 감소시간간격)이 지나면 1씩 감소하는 카운터를 도입하여, 이 카운터 값이 임계치 이상이 될 경우 과부하 상태가 시작되는 것으로 판단하고 과부하 제어 알고리즘이 동작한다. 과부하 제어가 시작되면 프로세서의 점유율을 측정하여 목표치를 넘거나 모자라는 비율대로 호카운터 감소 시간간격 값을 증감시키게 되며, 과부하 상태에서의 호 수락여부는 카운터 값이 특정값 이하일 경우에만 호를 수락하도록 하여 결국 과부하가 발생하면 허용되는 호의 수를 줄이게 된다. 과부하제어의 해제는 카운터 감소시간간격 값이 일정값보다 작아지면 과부하제어를 해제하도록 하였다. 또한 ATM 교환기의 특성인 셀폭주를 제어하기 위해 만일 스위치의 공통 메모리나 AFIFO 메모리의 셀수가 임계치 이상이 되면 셀폭주 제어를 시작하며, 셀폭주가 발생한 시간이 오래될수록 새로운 입력 호에 대한 수락확율을 지속적으로 작게되도록 조절하여 셀폭주를 방지한다.

본 논문에서 제안한 알고리즘을 검증하기 위해 시뮬레이션 모델을 개발하고, 먼저 호처리 프로세서의 과부하시에 동작을 검증하기 위해 이 모델에 다양한 입력 패턴을 인가하여 시뮬레이션 동안의 프로세서 점유율의 변화 등을 분석하였으며, 그 결과 프로세서 점유율이 목표 점유율에 항상 근접하게 유지됨을 확인하였다. 또한 셀폭주시의 알고리즘 동작을 검증하기 위해 호입력량이 각각 70%, 150%일 때 입력되는 셀 트래픽양을 변화시키면서 시뮬레이션한 결과 새로운 호 요구에 대한 수락확율을 적절히 조절함으로써 셀폭주에 의한 셀손실율을 줄임으로써 스위치계 과부하를 제어할 수 있음을 보였다.

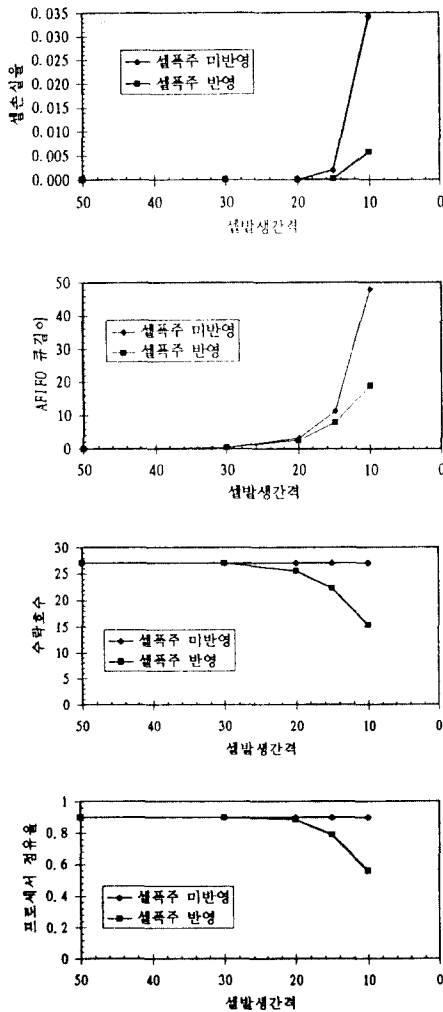


그림 9. 셀폭주에 대한 시뮬레이션 결과(호입력 150%)  
 Fig. 9 Simulation results for cell overflow(call input load 150%)



참고 문헌

1. TDX-10 총서, 제11권 성능평가, 한국전자통신연구소, TDX 개발단, 1991. 6.
2. 주성순, "축적 프로그램제어교환기의 과부하제어 방식에 관한 연구", 서울대학교 대학원 박사학위논문, 1988. 12.
3. C. H. Yim, H. L. Hartmann, "Throughput behavior of switching systems under heavy load conditions", 11th ITC, 1985.
4. Ulf K., "Overload control of SPC systems", 13th ITC, 1991.
5. Georg Daisenberger, J. Oehlerich, Gerhard Wegmann, "STATOR-statistical overload regulation-and tail-time account input limitation-two concepts for overload regulation in SPC systems", 11th ITC, pp. 120-126, 1985.
6. 최태암, 김정일, "No.1 A 전자교환기 발신호 과부하 처리 전략에 관한 고찰", 전기통신연구 제1권 제3호, pp.87-91, Oct., 1987.
7. 진정학, 김영호, "AXE-10에서의 부하제어기능", 전기통신연구 제1권 제2호, pp.127-131, Aug., 1987.
8. G.M.Andres, M.V.Altamirano, "System 12 Traffic overload control", Electrical communication Vol. 59, No.1/2, 1985.
9. 김용훈, 김연태, 서두수, "5ESS의 과부하 제어전략에 관한 고찰", 전기통신연구 제5권 제2호, pp. 90-96, July, 1991.
10. H.Lee, K.H.Kook, S.K.Lim, J.H.Baek, "Performance analysis of distributed control switching system in traffic overload environments", SICON 1991 Singapore, Sep., 1991.
11. 임석구, 국광호, 이재섭, 김영시, "분산구조 교환기에서의 과부하 제어", 전자통신 제14권 제 3호, 1992. 10.
12. 정현필, 임석구, 이윤현, "분산구조형 교환시스템의 과부하 제어", 한국통신학회논문지 Vol.19, No.8, pp.1578-1586, 1994. 8.

13. 김승희, 정지훈, 조재균, 김영선, 강석열, "ATM 교환시스템에서의 호 제어에 관한 연구", 한국통신학회 93년 하계종합학술발표회논문집, pp. 225-229, 1993. 7.



崔眞圭(Jin-Kyu Choe) 정회원  
 1958년 9월 20일생  
 1980년 2월:고려대학교 전자공학과(공학사)  
 1982년 2월:고려대학교 대학원 전자공학과(공학석사)  
 1987년 8월:고려대학교 대학원 전자공학과(공학박사)  
 1987년 9월~1990년 8월:대전산업대학교 전자계산학과 조교수

1990년 9월~현재:한남대학교 전자공학과 부교수  
 ※주관심분야: ATM 교환시스템, Interconnection network, 병렬처리 등.

金煥善(Young-Sun Kim) 정회원  
 1982년~현재:한국전자통신연구소 책임연구원  
 한국통신학회 논문지 제21권 제3호 참조



奇長根(Jang-Geun Ki) 정회원  
 1961년 7월 15일생  
 1986년 2월:고려대학교 전자공학과 졸업(공학사)  
 1988년 2월:고려대학교 대학원 전자공학과 졸업(공학석사)  
 1992년 2월:고려대학교 대학원 전자공학과 졸업(공학박사)

1992년 3월~현재:공주대학교 전자공학과 조교수  
 ※주관심분야: 컴퓨터 통신 및 프로토콜